2m 111 362

PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : H04L 12/58, G06F 17/60 — A1

(11) International Publication Number: WO 99/29073

(43) International Publication Date: 10 June 1999 (10.06.99)

(21) International Application Number: PCT/JP98/05381

(22) International Filing Date: 30 November 1998 (30.11.98)

(30) Priority Data:
9/347293 3 December 1997 (03.12.97) JP

(71) Applicant *(for all designated States except US)*: CASIO COMPUTER CO., LTD. [JP/JP]; 6–2, Hon–machi 1–chome, Shibuya–ku, Tokyo 151–8543 (JP).

(72) Inventors; and

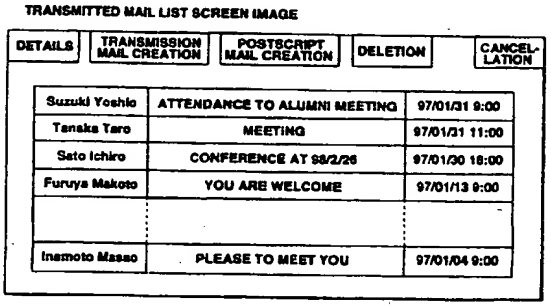
(75) Inventors/Applicants *(for US only)*: EGUCHI, Hirofumi [JP/JP]; 2–28–40, Wakamatsucho, Fuchu–shi, Tokyo 183–0005 (JP). YAMAGUCHI, Yoshito [JP/JP]; 1–17–17–702, Nishikicho, Tachikawa–shi, Tokyo 190–0022 (JP).

(74) Agents: SUZUYE, Takehiko et al.; Suzuye & Suzuye, 7–2, Kasumigaseki 3–chome, Chiyoda–ku, Tokyo 100–0013 (JP).

(81) Designated States: CN, KR, US, European patent (DE, FR, GB).

Published
With international search report.
Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(54) Title: ELECTRONIC MAIL APPARATUS AND STORAGE MEDIUM STORING ELECTRONIC MAIL PROGRAM



(57) Abstract

In addition to an ordinary mail including a receiver address, a subject and a text, a postcript mail which is same as a previously transmitted mail with respect to at least one of a receiver address and subject of the previously transmitted mail can be created and transmitted. When a received mail is the postscript mail and a mail that has a predetermined relation with the postscript mail is in the sealed condition, the mail and the postscript mail are stored in an integrated manner. When the postscript mail is in the unsealed condition, the postscript mail is stored as an independent mail.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	**ES**	Spain	**LS**	Lesotho	**SI**	Slovenia
AM	Armenia	**FI**	Finland	**LT**	Lithuania	**SK**	Slovakia
AT	Austria	**FR**	France	**LU**	Luxembourg	**SN**	Senegal
AU	Australia	**GA**	Gabon	**LV**	Latvia	**SZ**	Swaziland
AZ	Azerbaijan	**GB**	United Kingdom	**MC**	Monaco	**TD**	Chad
BA	Bosnia and Herzegovina	**GE**	Georgia	**MD**	Republic of Moldova	**TG**	Togo
BB	Barbados	**GH**	Ghana	**MG**	Madagascar	**TJ**	Tajikistan
BE	Belgium	**GN**	Guinea	**MK**	The former Yugoslav Republic of Macedonia	**TM**	Turkmenistan
BF	Burkina Faso	**GR**	Greece			**TR**	Turkey
BG	Bulgaria	**HU**	Hungary	**ML**	Mali	**TT**	Trinidad and Tobago
BJ	Benin	**IE**	Ireland	**MN**	Mongolia	**UA**	Ukraine
BR	Brazil	**IL**	Israel	**MR**	Mauritania	**UG**	Uganda
BY	Belarus	**IS**	Iceland	**MW**	Malawi	**US**	United States of America
CA	Canada	**IT**	Italy	**MX**	Mexico	**UZ**	Uzbekistan
CF	Central African Republic	**JP**	Japan	**NE**	Niger	**VN**	Viet Nam
CG	Congo	**KE**	Kenya	**NL**	Netherlands	**YU**	Yugoslavia
CH	Switzerland	**KG**	Kyrgyzstan	**NO**	Norway	**ZW**	Zimbabwe
CI	Côte d'Ivoire	**KP**	Democratic People's Republic of Korea	**NZ**	New Zealand		
CM	Cameroon			**PL**	Poland		
CN	China	**KR**	Republic of Korea	**PT**	Portugal		
CU	Cuba	**KZ**	Kazakstan	**RO**	Romania		
CZ	Czech Republic	**LC**	Saint Lucia	**RU**	Russian Federation		
DE	Germany	**LI**	Liechtenstein	**SD**	Sudan		
DK	Denmark	**LK**	Sri Lanka	**SE**	Sweden		
EE	Estonia	**LR**	Liberia	**SG**	Singapore		

DESCRIPTION

ELECTRONIC MAIL APPARATUS AND STORAGE MEDIUM STORING ELECTRONIC MAIL PROGRAM

Technical Field

The present invention relates to an electronic mail apparatus for transmission and reception of an electronic mail and a storage medium storing an electronic mail program.

Background Art

In a conventional electronic mail apparatus, there has been conducted procedures that when a transmitter prepares a mail for transmission, an address and a subject are first set and thereafter, a text is drafted and thus created mail is transmitted, while a receiver can read a content of the mail by displaying the content thereof after unsealing the mail.

However, when a transmitter has sent a first mail with a subject and thereafter wishes to add a paragraph or a phrase to a content of the transmitted first mail, a second mail has to be created and transmitted so as to include a description to the effect that the second mail is related to the first mail as well as the paragraph or the phrase to be added to the first mail, which makes mail creation work unavoidably complicated.

On the receiver side, however, when the receiver has not unsealed the first mail yet, the receiver has

to unseal the two mails. In addition, when the second mail is transmitted with an interval after the first mail is transmitted, there is a risk that the two mails are not recognized as ones which are mutually related to each other.

Disclosure of Invention

An object of the present invention is to provide an electronic mail apparatus by which a mail having a paragraph or a phrase to be added to a mail that has previously been transmitted can be created and transmitted with a simple operation.

According to a first aspect of the present invention, there is provided an electric mail apparatus comprising:

ordinary mail creating means for creating an ordinary mail including a receiver address, a subject and a text;

ordinary mail transmitting means for transmitting the ordinary mail created by the ordinary mail creating means;

transmitted mail storage means for storing an ordinary mail transmitted by the ordinary mail transmitting means;

transmitted mail designating means for designating one of transmitted mails stored in the transmitted mail storage means;

postscript mail creating means for creating a

postscript mail which is same as the mail designated by the transmitted mail designating means with respect to at least one of a receiver address and subject of the designated mail;

postscript mail transmitting means for transmitting the postscript mail created by the postscript mail creating means and storing a transmitted postscript mail in the transmitted mail storage means;

mail receiving means for receiving a mail;

received mail storage means for storing the mail received by the mail receiving means;

received mail designating means for designating one of received mails stored in the received mail storage means; and

unsealing means for displaying at least one of a transmitter address, subject and text of the received mail designated by the received mail designating means.

According to the first aspect of the present invention, a postscript mail for a previously transmitted mail can be transmitted by a simple operation.

Another object of the present invention is to provide an electronic mail apparatus which can make simple works in mail creation on the transmitter and in unsealing on the receiver when a mail which adds a paragraph or a phrase to a content of a previously

transmitted mail is later transmitted.

According to a second aspect of the present invention, there is provided an electronic mail apparatus comprising:

mail creating means for creating an ordinary mail including a receiver address, a subject and a text and a postscript mail having a predetermined relation with a previously transmitted mail;

mail transmitting means for transmitting the mail created by the mail creating means;

mail receiving means for receiving a mail;

mail storage means for storing the mail received by the mail receiving means;

unsealing means for designating one of the received mails stored in the mail storage means and displaying at least one of a transmitter address, subject and text of the designated mail;

specific mail detecting means for detecting whether or not the mail received by the mail receiving means is a postscript mail;

mail extracting means for extracting a mail which has a predetermined relation with the postscript mail among the received mails stored in the mail storage means when the mail detected by the postscript mail detecting means is the postscript mail;

unseal detecting means for detecting whether or not the mail extracted by the mail extracting means has

been unsealed by the unsealing means; and

storage control means for integrating the extracted mail and the received mail with relating to each other in the mail storage means when the extracted mail is detected as having not been unsealed by the unseal detecting means, and storing the received mail in the mail storage means when the mail detected by the postscript mail detecting means is not the postscript mail or when the extracted mail is detected as having been unsealed by the unseal detecting means.

According to the second aspect of the present invention, when a mail by which a paragraph or a phrase is added to a content of a previously transmitted mail is transmitted after the previous mail is transmitted, works in mail creation on the transmitter and in unsealing on the receiver can be reduced to be simple.

A further object of the present invention is to provide a storing medium for storing a program for an electronic mail, the program making it possible to prepare and transmit a postscript mail for a previously transmitted mail by a simple operation.

According to a third aspect of the present invention, there is provided a storage medium readable by a computer storing a program for electronic mail processing, the program comprising:

a program code A for creating an ordinary mail including a receiver address, a subject and a text;

a program code B for transmitting the ordinary mail created by the program code A;

a program code C for storing the ordinary mail transmitted by the program code B;

a program code D for designating one of transmitted mails stored by the program code C;

a program code E for creating a postscript mail which is same as the mail designated by the program code D with respect to at least one of a receiver address and subject of the designated mail;

a program code F for transmitting the postscript mail created by the program code E and storing a transmitted postscript mail;

a program code G for receiving a mail;

a program code H for storing a mail received by the program code G;

a program code I for designating one of received mails stored by the program code H; and

a program code J for displaying at least one of a transmitter address, subject and text of the received mail designated by the program code I.

According to the third aspect of the present invention, it becomes possible to transmit a postscript mail for a previously transmitted mail by a simple operation.

Still another object of the present invention is to provide a storing medium for storing a program for

an electric mail which can make simple works in mail creation on the transmitter and in unsealing on the receiver side, when a mail by which a paragraph or a phrase is added to a previously transmitted mail is transmitted after the previous mail is transmitted.

According to a fourth aspect of the present invention, there is provided a storage medium readable by a computer storing a program for electronic mail processing, the program comprising:

a program code A for creating an ordinary mail including a receiver address, a subject and a text but also a postscript mail having a predetermined relation with a previously transmitted mail;

a program code B for transmitting the mail created by the program code A;

a program code C for receiving a mail;

a program code D for storing the mail received by the program code C;

a program code E for designating one of the received mails stored by the program code D and displaying at least one of a transmitter address, subject and text of the designated mail;

a program code F for detecting whether or not the mail received by the program code C is a postscript mail;

a program code G for extracting a mail which has a predetermined relation with the postscript mail among

the received mails stored by the program code D when the mail detected by the program code F is the postscript mail;

a program code H for detecting whether or not the mail extracted by the program code G has been unsealed; and

a program code I for integrating the extracted mail and the received mail with relating to each other when the extracted mail is detected as having not been unsealed by the program code H, and storing the received mail when the mail detected by the program code F is not the specific mail or when the extracted mail is detected as having been unsealed by the program code H.

According to the fourth aspect of the present invention, when a mail by which a paragraph or a phrase is added to a content of a previously transmitted mail is transmitted after the previous mail is transmitted, works in mail creation on the transmitter and in unsealing on the receiver can be reduced to be simple.

Additional objects and advantages of the present invention will be set forth in the description which follows, and in part will be obvious from the description, or may be learned by practice of the present invention.

The objects and advantages of the present invention may be realized and obtained by means of the

instrumentalities and combinations particularly pointed out hereinafter.

Brief Description of Drawings

The accompanying drawings, which are incorporated in and constitute a part of the specification, illustrate presently preferred embodiments of the present invention and, together with the general description given above and the detailed description of the preferred embodiments given below, serve to explain the principles of the present invention in which:

FIG. 1 is a block diagram showing a system structure of an electronic mail apparatus according to an embodiment of the present invention;

FIG. 2 shows a memory map, which is a data structure, of RAM in FIG. 1;

FIG. 3 is a flowchart for a main routine of CPU of FIG. 1;

FIG. 4 is a flowchart for a received mail list display processing at step S2 of FIG. 3;

FIG. 5 is a flowchart for the received mail list display processing, continued to FIG. 4;

FIG. 6 is a display view showing a screen image of a received mail list;

FIG. 7 is a flowchart for a transmitted mail list display processing at step S34 of FIG. 5;

FIG. 8 is a flowchart for the transmitted mail list display processing, continued to FIG. 7;

FIG. 9 is a flowchart for the transmitted mail list display processing, continued to FIG. 8;

FIG. 10 is a display view showing a screen image of a transmitted mail list;

FIG. 11 is a flowchart for unseal processing at step S30 of FIG. 5;

FIG. 12 is a display view showing an unsealing screen image of a received mail;

FIG. 13 is a flowchart for detailed display processing at step S68 of FIG. 9;

FIG. 14 is a display view showing a detailed display screen image of a received mail;

FIG. 15 is a flowchart for mail creation processing at step S32 of FIG. 5;

FIG. 16 is the flowchart for mail creation processing, continued to FIG. 15;

FIG. 17 is the flowchart for mail creation processing, continued to FIG. 16;

FIG. 18 is the flowchart for mail creation processing, continued to FIG. 15;

FIG. 19 is a display view showing a mail creation screen image;

FIG. 20 is a flowchart for address list display processing at step S111 of FIG. 16;

FIG. 21 is a display view showing an address list display screen image;

FIG. 22 is a flowchart for address input

processing in FIGS. 5 and 20;

FIG. 23 is the flowchart for address input processing, continued to FIG. 22;

FIG. 24 is a display view showing an address input screen image;

FIG. 25 is a flowchart for postscript mail creation processing at step S93 of FIG. 13;

FIG. 26 is the flowchart for postscript mail creation processing, continued to FIG. 25;

FIG. 27 is a display view showing a postscript mail creation screen image;

FIG. 28 is a flowchart for timer interrupt processing;

FIG. 29 is a display view showing an integrated mail unseal screen image; and

FIG. 30 is a display view showing a postscript mail unseal screen image.

Best Mode of Carrying Out the Invention

FIG. 1 is a block diagram showing a system structure of an electronic mail apparatus according to an embodiment of the present invention. A plurality of terminals 100 of an electronic mail apparatus are respectively connected to a server 300 through a dedicated line 200 all the time. Each terminal 100 accesses to the server 300 responsively to timer interrupt occurring at given intervals to inquiry about whether or not an electronic mail addressed to the

terminal 100 has reached and, when an electronic mail has reached, receives the electronic mail from the server 300.

In each terminal 100, CPU 1 is connected to not only a dedicated line 200 through a system bus 101 but also internally to ROM 2, RAM 3, a keyboard 4, a display device 5 and a mouse 6 through the system bus 101. ROM 2 stores programs which CPU 1 executes and initial data in initialization processing. RAM 3 stores data of an electronic mail for transmission which is input from the keyboard 4 and data of a received electronic mail from the server 300. Therefore, in RAM 3, there are provided, as shown FIG. 2, a mail creation area 31, a transmitted mail area 32, a received mail area 33 and an address area 34.

Data of the mail creation area 31 comprise a postscript flag, a transmitter address, a receiver address, a subject and a text. The transmitted mail area 32 comprises areas of MAIL(1) to MAIL(N) which enables N mails to be stored. Data of each MAIL(1) to MAIL(N) areas comprises a postscript flag, a transmitter address, a receiver address, a subject, a text and a reception date/time. The received mail storage area 33 comprises areas of MAIL(1) to MAIL(M) which enables M mails to be stored. Data of each MAIL(1) to MAIL(M) areas comprises a postscript flag, a transmitter address, a receiver address, a subject, a

text, a reception date/time and an unseal flag. An unseal flag is set to "1" when a received electronic mail has been unsealed, and to "0" when a received electronic mail has not been unsealed. In the address area 34, registered are addresses of a plurality of other terminals 100, each of which is to be a transmitter, which consist of L addresses of ADDRESS(1) to ADDRESS(L). Each address comprises a name and mail address of a transmitter.

In FIG. 1, the keyboard 4 is provided with a variety of switches (not shown). A mail switch is a switch for transition to mail processing. A character-input switch and a ten key switch are switches for inputting a mail address or a transmission sentence. The display device 5 displays a mail to be transmitted and a received mail. The mouse 6 moves a cursor on the screen image of the display device 5 in a corresponding manner to its operation and not only designates a position of character inputting from the key board 4 but also performs various selections by clicking a click switch.

Then, there will be described operations in an embodiment in reference to a flow chart to be executed by CPU 1 and a screen image displayed on the display device 5.

FIG. 3 is a flowchart for a main routine of CPU of FIG. 1. At step S1, it is determined whether or not a

mail mode flag MMF, which indicates processing of a received mail, is "1" indicating a mail processing. When this flag is "1", display processing of a received mail list is performed (step S2) and when this flag is "0", one of other processings than the mail processing is executed (step S3).

After the processing at steps S2 or S3, the keyboard 4 is scanned to determine whether or not a switch is turned on (step S4). When a switch has been turned on, it is determined whether or not the turned-on switch is a mail switch to issue an instruction for mail processing (step S5). When the mail switch has been turned on, the flag MMF is set to "1" (step S6). After the setting of MMF, or when none of the switches is turned on at step 4 or when the switch which has been turned on at step S5 is not the mail switch, program flow returns to step S1 and the above described processing is repeatedly executed.

FIGS. 4 and 5 are flows for a received mail list display processing at step S2 in FIG. 3 and FIG. 6 is a flow for a received mail list screen image. In FIG. 4, the display is cleared (step S10) and a frame screen image of a received mail is displayed (step S11). Then, pointer "n" which designates the number of MAIL(n) in the received mail area 33 of RAM 3 is set to "1" (step S12) and the following loop processing is executed as "n" is incremented. That is, MAIL(n) designated by "n"

is displayed (step S13) and it is determined whether or not an unseal flag KAIFUF(n) of the mail is "0" (step S14). When this flag is "0", since the received mail is in the sealed condition, a unseal mark ◎ is displayed in a space on the left of a name space in the received mail list of FIG. 6 (step S15). At step S14, when KAIFUF(n) is "1", since the received mail has been unsealed, an unseal mark is not displayed.

Then, "n" is incremented (step S16) and it is determined whether or not "n" is equal to or less than the maximum number (step S17). When "n" is equal to or less than the maximum number, then it is determined whether or not MAIL(n) is free (step S18). When MAIL(n) is not free, program flow goes to step S13 and MAIL(n) is displayed on the screen. A loop from step S13 to step S18 is repeated till MAIL(n) is free and received mails are sequentially displayed.

At step S17, when pointer "n" is larger than the maximum number or at step S18 when MAIL(n) designated by "n" is free, register "A" which designates a received mail to be displayed in reverse is set to "1" (step S19) and in a screen image of FIG. 6, MAIL(A) is displayed in reverse (step S20). Therefore, in the first place, a received mail at the leading position of the received mail list is displayed in reverse.

Then, a cursor is displayed on the screen image (step S21) and it is determined whether or not the

mouse 6 has been moved (step S22). When the mouse 6 has been moved, a cursor display position is moved in a corresponding manner to the movement (step S23). Then, it is determined whether or not the mouse 6 has been clicked (step S24). When the mouse 6 has not been clicked, then program flow goes to step S22 and it is determined whether or not the mouse 6 has been moved. When the mouse 6 has been clicked, it is determined whether or not a cursor position is present in a position of a received mail list on the screen image (step S25). When the cursor position is a position of one of received mails, pointer "A" is set to the mail number of a cursor position (step S26). Then, MAIL(A) is displayed in reverse (step S27). The mail list excluding MAIL(A) is displayed in the normal condition (step S28). Thereafter, program flow goes to step S22 and it is determined whether or not the mouse 6 has been moved.

At step S25, when a cursor position is not present on the mail list screen image, processing is advanced to the flow of FIG. 5 and it is determined whether or not the cursor position is present in a position of one of the unseal switch, mail creation switch, transmitted mail list switch, address input switch, deletion switch and cancellation switch among icon switches disposed in the upper portion of the screen image of FIG. 6.

That is, it is determined whether or not a cursor

position is present in a position of the unseal switch (step S29) and when the cursor position is this position, unseal processing is executed (step S30). Thereafter, program flow goes to step S11 of FIG. 4 and a frame screen image of a received mail is displayed. When the cursor position is not present in an unseal switch position, it is determined whether or not a cursor position is a position of the mail creation switch (step S31) and when the cursor position is this position, the mail creation processing is executed (step S32). Thereafter, program flow returns to step S11 of FIG. 4 and a frame screen image of a received mail is displayed. When a cursor position is not present in a position of the mail creation switch either, it is determined whether or not the cursor position is a position of the transmitted mail list switch (step S33) and when the cursor position is present in this position, transmitted mail list display processing is executed (step S34). Thereafter, program flow returns to step S11 of FIG. 4 and a frame screen image of a received mail is displayed.

When a cursor position is not present in the position of the transmitted mail list switch either, it is determined whether or not the cursor position is present in a position of the address input switch (step S35) and when the cursor position is present in this position, address input processing is executed (step

S36). Thereafter, program flow returns to step S11 of FIG. 4 and a frame screen image of a received mail is displayed. When a cursor position is not present in the position of the address input switch either, it is determined whether or not the cursor position is present in a position of the deletion switch (step S37) and the cursor position is present in this position, MAIL(A) is deleted from RAM 3 (step S38) and KAIFUF(A) is reset to 0 (step S39). In addition, received mails are sorted in the order of a reception time (step S40). Thereafter, program flow returns to step S11 of FIG. 4 and a frame screen image of a received mail is displayed in order to display all the received mails except a received mail which has been deleted.

At step S37 of FIG. 5, when a cursor position is not present in the deletion switch either, it is determined whether or not the cursor position is present in a position of the cancellation switch (step S41) and when the cursor position is present in this position, the display of the received mail list screen image is cleared (step S42) and a flag MMF is reset to "0" (step S43). Then, the received mail list display processing is terminated. When a cursor position is not present in a position of the cancellation switch either, program flow returns to step S22 of FIG. 4 and it is determined whether or not the mouse 6 has been moved.

FIGS. 7 to 9 are flows for transmitted mail list display processing at step S34 of FIG. 5 and FIG. 10 is a screen image of a transmitted mail list.

In FIG. 7, the display is cleared (step S44) and a frame screen image of a transmitted mail is displayed (step S45). Then, pointer "n" which designates the number of MAIL(n) in the transmitted mail area 32 of RAM 3 is set to "1" (step S46) and the following loop is executed as "n" is incremented. That is, MAIL(n) designated by "n" is displayed on the screen image of FIG. 10 (step S47). Then, "n" is incremented (step S48) and it is determined whether or not "n" is equal to or less than the maximum number (step S49). When "n" is equal to or less than the maximum number, it is determined whether or not MAIL(n) area is free (step S50). When MAIL(n) area is not free, program flow returns to step S47 and MAIL(n) is displayed on the screen image. Then, a loop from step S47 to step 50 is repeated till MAIL(n) area is free and transmitted mails are sequentially displayed.

At step S49, when pointer "n" is larger than the maximum number, or at step S50 when MAIL(n) designated by "n" is free, register "A" which designates a transmitted mail to be displayed in reverse is set to "1" (step S51) and MAIL(A) is displayed in reverse on the screen image of FIG. 10 (step S52). Therefore, a received mail at the leading position of the

transmitted mail list is displayed in reverse in the first place.

Then, in a flow of FIG. 8, a cursor display is presented in the screen image of FIG. 10 (step S53) and it is determined whether or not the mouse 6 has been moved (step S54). When the mouse 6 has been moved, a cursor position is moved in a corresponding manner to the movement of the mouse 6 (step S55). Then, it is determined whether or not the mouse 6 has been clicked (step S56). When the mouse 6 has not been clicked, program flow returns to step S54 and it is determined whether or not the mouse 6 has been moved. When the mouse 6 has been clicked, it is determined whether or not a cursor position then is present in a position of a screen image of the transmitted mail list (step S57). When a cursor position is present in a position of one of transmitted mails, pointer "A" is set to a mail number of the cursor position (step S58). Then, MAIL(A) is displayed in reverse (step S59). The mail list other than MAIL(A) which are displayed in reverse is changed to be in the normal display condition (step S60). Thereafter, program flow goes to step S54 and it is determined whether or not the mouse has been moved.

At step S57, when a cursor position is not present on the screen image of a mail list, it is determined whether or not the cursor position is present in a position of one of a details switch, a received mail

list switch, a postscript creation switch, deletion switch and a cancellation switch of the icon switches in the upper portion of the screen image of FIG. 10.

That is, it is determined whether or not a cursor position is present in a position of the deletion switch (step S61) and when the cursor position is present in this position, MAIL(A) is deleted from RAM 3 (step S62) and mails in the transmitted mail area are sorted (step S63). Thereafter, program flow returns to step S45 of FIG. 7 and a frame screen image of a transmitted mail is displayed in order to display all the transmitted mails except a transmitted mail which has been deleted.

At step S61 of FIG. 8, when a cursor position is not present in a position of the deletion switch, it is determined whether or not the cursor position is a position of the cancellation switch (step S64) and when the cursor position is present in this position, the display of the received mail list screen image of FIG. 10 is cleared (step S65) and a flag MMF is reset to "0" (step S66). Then, the transmitted mail list display processing is terminated.

At step S64, when a cursor position in not present in a position of the cancellation switch either, in the flow of FIG. 9, it is determined whether or not the cursor position is present in a position of the details switch (step S67) and when the cursor position is

present in this position, detailed display processing is executed (step S68). When a cursor position is not present in a position of the details switch either, it is determined whether or not the cursor position is present in a position of the postscript mail creation switch (step S69) and when the cursor position is present in this position, postscript mail creation processing is executed (step S70). When a cursor position is not present in a position of the postscript mail creation switch either, it is determined whether or not the cursor position is present in a position of the received mail list switch (step S71) and when the cursor position is present in this position, received mail list display processing in FIG. 4 is executed (step S72).

After the detailed display processing at step S68, after the postscript mail creation processing at step S70 or after the received mail list display processing at steps S72, program flow returns to step S45 of FIG. 7 and a frame screen image of the transmitted mail list of FIG. 10 is displayed.

FIG. 11 is a flow for unseal processing of a received mail at step S30 of FIG. 5 and FIG. 12 is an unseal screen image of a received mail.

In FIG. 11, at first, a mail display frame screen image is displayed (step S73). Then, an address of MAIL(A) which is in a reverse display condition is

presented in a receiver address in the display frame screen image (step S74), subject data of MAIL(A) is displayed in the subject area (step S75), text data of MAIL(A) is displayed in the text area (step S76). In response to the unseal instruction, a unseal flag KAIFUF(A) is set to "1" (step S77).

Then, a cursor is displayed on a screen image of FIG. 12 (step S78) and it is determined whether or not the mouse 6 has been moved (step S79). When the mouse 6 has been moved, the cursor is moved in a corresponding manner to the movement (step S80). Then, it is determined whether or not the mouse 6 has been clicked (step S81). When the mouse 6 has been clicked, it is determined whether or not a cursor position is present in a position of the cancellation switch (step S82) and when the cursor position is present in this position, the display of a received mail unseal screen image of FIG. 12 is cleared (step S83) and this unseal processing is terminated. At step S81, when the mouse has not been clicked, or at step S82 when a cursor position is not present in a position of the cancellation switch, program flow returns to step S79 and it is determined whether or not the mouse 6 has been moved.

FIG. 13 is a flow for the details processing at step S68 of FIG. 9 and FIG. 14 is a detailed display screen image.

In FIG. 13, a frame screen image of a mail display is displayed (step S84) and a transmitter address, a receiver address, a text of MAIL(A), which has been displayed in reverse at step S59 of FIG. 8, is displayed (step S85). Then, a cursor is displayed on the screen image of FIG. 14 (step S86). Then, it is determined whether or not the mouse 6 has been moved (step S87) and when the mouse 6 has been moved, a cursor display position is moved in a corresponding manner to the movement (step S88). Then, it is determined whether or not the mouse 6 has been clicked (step S89) and when the mouse 6 has been clicked, it is determined whether or not a cursor position is in a position of the cancellation switch (step S90). When the cursor position is present in this position, the display on the screen image of FIG. 14 is cleared (step S91) and this detailed display processing is terminated.

At step S90, when a cursor position is not in a position of the cancellation switch, it is determined whether or not the cursor position is present in a position of the postscript switch (step S92) and when the cursor position is present in this position, postscript mail creation processing is executed (step S93). Then, the detailed display processing is terminated.

FIGS. 15 to 18 are flows for the mail creation processing at step S32 of FIG. 5 and FIG. 19 is a mail

creation screen image.

In FIG. 15, the mail creation area 31 of RAM 3 is cleared (step S94) and a postscript flag of the mail creation area is reset to "0" (step S95). Then, a transmitter address in the mail creation area is used as a self ADDRESS (step S96). Then, a mail creation screen image shown in FIG. 19 is displayed based on the mail creation area (step S97). A character cursor is displayed in the address-input area of the screen image (step S98) and a cursor is displayed on the screen image (step S99).

It is determined whether or not the mouse 6 has been moved (step S100) and when the mouse 6 has been moved, a cursor position is moved in a corresponding manner to the movement (step S101). Then, it is determined whether or not the mouse 6 has been clicked (step S102) and when the mouse 6 has been clicked, it is determined whether or not a cursor position is present in a position of the input area in the flow of FIG. 16 (step S103). When the cursor position is present in this position, a character cursor is moved in a corresponding manner to the cursor position. That is, it is determined whether or not a cursor position is present in a position of the address input area (step S104) and when the cursor position is present in this position, the character cursor position is moved to the address input area (step S105). It is

determined whether or not a cursor position is present in a position of the subject input area (step S106) and when the cursor position is present in this position, the character cursor is moved to the subject input area (step S107). It is determined whether or not a cursor position is present in a position of the text input area (step S108) and when the cursor position is present in this position, a character cursor is moved to the text input area (step S109).

When a cursor position is not present in the text input area, it is determined whether or not a cursor position is present in a position of the address list switch (step S110) and when the cursor position is present in this position, address display processing is executed (step S111). Then, program flow returns to step S97 of FIG. 15 and a mail creation screen image is displayed. At step S110 of FIG. 16, when a cursor position is not present in a position of the address list switch either, it is determined in the flow of FIG. 17 whether or not a cursor position is present in a position of the transmitted switch (step S112). When the cursor position is present in this position, transmission processing is executed (step S113).

At step S112, when a cursor position is not present in a position of the transmission switch either, it is determined whether or not the cursor position is in a position of the cancellation switch (step S114).

When the cursor position is present in this position, or after the transmission processing at step S113, pointer "m" which designates a transmitted mail is set to "1" (step S115) and it is determined whether or not MAIL(m) in the transmitted mail area is free, as "m" is incremented (step S116).

When MAIL(m) area is not free, "m" is incremented (step S117) and it is determined whether or not "m" has exceeded the maximum number (step S118). When "m" is equal to or less than the maximum number, program flow returns to step S116 and it is determined whether or not MAIL(m) area is free. Then, processing from step S116 to step S118 is repeated to search for a free area in the transmitted mail area. At step S118, when "m" has exceeded the maximum number, the number of a transmitted mail with the earliest transmission date/time is set to "m" (step S119).

At step S116, when an area of MAIL(m) is free, or when a mail number is set to "m" at step S119, data in the mail creation area is stored in MAIL(m) (step S120). Then, the transmission date/time of MAIL(m) is set to the present date/time (step S121). Then, the display is cleared (step S122), the mail creation area is cleared (step S123) and this mail creation processing is terminated. At step S114, when a cursor position is not present in a position of the cancellation switch, program flow goes to step S100 of FIG. 15 and it is

determined whether or not the mouse 6 has been moved.

At step S102 of FIG. 15, when the mouse 6 has not been clicked, it is determined in a flow of FIG. 18 whether or not data input has been made from the keyboard 4 (step S124). When data input has not been made, program flow returns to step S100 of FIG. 15 and it is determined whether or not the mouse 6 has been moved. When data input has been made, it is determined whether or not a character cursor position is present in a position of the address-input area (step S125). When the character cursor is present in this position, input data is displayed in the address-input area (step S126) and stored in an address area of a mail creation buffer of RAM 3 (step S127). Then, a character cursor is moved to a next character input position (step S128).

At step S125, when a character cursor position is not present in a position of the address-input area, it is determined whether or not the character cursor position is present in a position of the subject-input area (step S129). When the character cursor position is present in this position, input data is displayed in the subject-input area (step S130) and the input data is stored in a subject area of the mail creation buffer of RAM 3 (step S131). Then, a character cursor is moved to a next character input position (step S132).

At step S129, when a character cursor position is not present in a position of the subject-input area, it

is determined whether or not the character cursor position is present in a position of the text-input area (step S133). When the character cursor position is present in this position, input data is displayed in the text-input area (step S134) and the input data is stored in a text area of the mail creation buffer of RAM 3 (step S135). Then, the character cursor is moved to a next character input position (step S136). After a character cursor is moved at step S128, step S132 or step S136, program flow returns to step S100 of FIG. 15 and it is determined whether or not the mouse 6 has been moved.

FIG. 20 is a flow for address list display processing at step S111 of FIG. 16 and FIG. 21 is an address list display screen image.

In FIG. 20, in the first place, a frame screen image of an address list is displayed (step S137). Then, pointer "m" indicating the address number is set to "1" (step S138) and data of ADDRESS(m) designated by "m" is displayed (step S 139). Then, "m" is incremented (step S140) and it is determined whether or not "m" has exceeded the maximum number (step S141). When "m" is equal to or less than the maximum number, program flow returns to step S139 and data of ADDRESS(m) is displayed. Then, a loop including step S139, step 140 and step 141 is repeated till "m" exceeds the maximum number.

At step S141, when "m" has exceeded the maximum number, register "A" is set to "1" (step S142A) and ADDRESS(A) of the leading position of the address list is displayed in reverse (step S142B). Then, a cursor is displayed on the screen image (step S142C) and it is determined whether or not the mouse 6 has been moved (step S143). When the mouse 6 has been moved, a cursor position is moved in a corresponding manner to the movement (step S144). Then, it is determined whether or not the mouse 6 has been clicked (step S145) and when the mouse 6 has not been clicked, program flow returns to step S143 and it is determined whether or not the mouse 6 has been moved.

When the mouse 6 has been clicked, it is determined whether or not a cursor position is present in a position of the address list on the screen image of FIG. 21 (step S146). When the cursor position is present in this position, "A" is set to the address number of the cursor position (step S147). Then, ADDRESS(A) is displayed in reverse (step S148) and the other addresses are displayed in the normal condition (step S149). Then, program flow returns to step S143 and it is determined whether or not the mouse 6 has been moved.

At step S146, when a cursor position is not present in a position of the address list, it is determined whether or not the cursor position is

present in a position of a new input switch on the screen image (step S150). When the cursor position is present in this position, address input processing is executed (step S151). After completion of the address input processing, program flow goes to step S137 and a screen image of the address list is displayed.

At step S150, when a cursor position is not present in a position of the new input switch, it is determined whether or not the cursor position is in a position of the cancellation switch on the screen image (step S152). When the cursor position is present in this position, the display of the address list screen image of FIG. 21 is cleared (step S153) and this address list display processing is terminated. At step S152, when a cursor position is not present in a position of the cancellation switch, it is determined whether or not a cursor position is present in a position of the deletion switch on the screen image (step S154). When the cursor position is present in this position, ADDRESS(A) is deleted from the address area of RAM 3 (step S155) and address data are sequentially sorted (step S156). Then, program flow returns to step S143 and it is determined whether or not the mouse 6 has been moved.

At step S154, when a cursor position is not present in a position of the deletion switch, it is determined whether or not the cursor position is

present in a position of the OK switch on the screen image (step S157). When the cursor position is present in this position, data of ADDRESS(A) is stored in the mail creation buffer (step S158) and the display of the address list screen image is cleared (step S159) and this mail list display processing is terminated.

FIGS. 22 and 23 are flows for the address input processing at step S36 of FIG. 5 and step S151 of FIG. 20 and FIG. 24 is an address input screen image.

In FIG. 22, pointer "m" which indicates an address number is set to "1" (step S161) and it is determined whether or not an area of ADDRESS(m) is free (step S162). When this area is not free, "m" is incremented (step S163) and it is determined whether or not "m" has exceeded the maximum number (step S164). When "m" is equal to or less than the maximum number, program flow returns to step S162 and it is determined whether or not an area of an ADDRESS(m) is free. Then, a loop including step S162, step S163 and step S164 is repeated to search a free area, as "m" is incremented. At step S164, when "m" has exceeded the maximum number, since there is available no free area, a warning display is presented for a given time period (step S165) and this address input processing is terminated.

At step S162, when ADDRESS(m) with a free area is present, an address input screen image shown in FIG. 24 is displayed (step S166). Not only is a cursor

displayed on the screen image (step S167) but also a character cursor is displayed in a name input area (step S168). Then, it is determined whether or not the mouse 6 has been moved (step S169) and when the mouse 6 has been moved, a cursor is moved in a corresponding manner to the movement (step S170). Then, it is determined whether or not the mouse 6 has been clicked (step S171) and when the mouse 6 has been clicked, it is determined whether or not a cursor position is present in a position of the OK switch on the screen image of FIG. 24 (step S172). When the cursor position is present in this position, the display of the address input screen image is cleared (step S173) and this address input processing is terminated.

At step S171, when a position where the mouse 6 has been clicked is not present in a position of the OK switch, it is determined whether or not a cursor position is present in a position of the cancellation switch on the screen image (step S174). When the cursor position is present in this position, a content of ADDRESS(m) is cleared (step S175) and the display of the address input screen image is cleared (step S173) and this address input processing is terminated.

At step S174, when a cursor position is not present in a position of the cancellation switch, it is determined in FIG. 23 whether or not the cursor position is present in a position of name (step S176).

When the cursor position is present in this position, a character cursor is move to the name input area (step S177). At step S176, when a cursor position is not present in a position of name, it is determined whether or not the cursor position is a position of the address-input area (step S178). When the cursor position is present in this position a character cursor is moved to the address-input area (step S179). When the cursor position is not in a position of the address-input area, or after a character position is moved at step S177 or step S179, program flow returns to step S169 of FIG. 22 and it is determined whether or not the mouse 6 has been moved.

At step S171 of FIG. 22, when the mouse 6 has not been clicked, it is determined in a flow of FIG. 23 whether or not data input from the keyboard 4 has been made (step S180). When the data input has been made, it is determined whether or not a character cursor is present in a position of name (step S181). When the character cursor position is present in this position, data input in the name input area is displayed (step S182) and the input data is stored in the name input area of ADDRESS(A) of RAM 3 (step S183). Then, the character cursor is moved (step S184).

At step S181, when a character cursor position is not present in a position of name, it is determined whether or not the character cursor position is present

in a position of the address input (step S185). When the character cursor position is present in this position, data input in the address input area is displayed (step S186) and the data input in the address area of ADDRESS(A) of RAM 3 is stored (step S187). Then, the character cursor position is moved (step S184). After the character cursor is moved, or when a character cursor position is not present in a position of the address input at step S185, program flow returns to step S169 of FIG. 22 and it is determined whether or not the mouse 6 has been moved.

FIGS. 25 and 26 are flows for the postscript mail creation processing at step S93 of FIG. 13 and FIG. 27 is a postscript mail creation screen image.

In FIG. 25, the mail creation area of RAM 3 is cleared (step S188) and a postscript flag of the mail creation area is set to "1" (step S189). Then, a transmitter address of the mail creation area is used as a self-ADDRESS (step S190). An address of the mail creation area is made same as an address of MAIL(A) which is displayed in reverse at step S59 of FIG. 8 (step S191). In addition, a subject of the mail creation area is made same as a subject of MAIL(A) (step S192).

Then, a string of characters of "Postscript: " and a mark are stored in the text of the mail creation area (step S193). Then, a postscript mail creation screen

image is displayed based on a content of the mail creation area (step S194). Then, a character cursor is displayed in the text area (step S195) and a cursor is displayed on the screen image (step S196). Then, it is determined whether or not the mouse 6 has been moved (step S197) and when the mouse 6 has been moved, a cursor position is moved in a corresponding manner to the movement (step S198). Then, it is determined whether or not the mouse 6 has been clicked (step S199) and when the mouse 6 has not been clicked, it is determined whether or not data input has been made from the keyboard 4 (step S200). When the data input has been made, the input data are stored in the text of the mail creation area (step S201) and the data is displayed in the text-input area on the screen image of FIG. 27 (step S202). Then, the character cursor is moved (step S203) and program flow goes to step S194 and the mail creation screen image is displayed. At step S200, when data has not been input, program flow goes to step S197 and it is determined whether or not the mouse 6 has been moved.

At step S199, when the mouse 6 has been clicked, it is determined in a flow of FIG. 26 whether or not a cursor position is present in a position of the transmission switch on the screen image of FIG. 27 (step S204). When the cursor position is present in this position, transmission processing is executed

(step S205). Then pointer "m" is set to "1" (step S206) and it is determined whether or not an area of MAIL(m) of the transmitted mail area of RAM 3 is free, as "m" is incremented (step S207).

When the area of MAIL(m) is not free, "m" is incremented (step S208) and it is determined whether or not "m" has exceeded the maximum number (step S209). When "m" is equal to or less than the maximum number, program flow goes to step S207 and it is determined whether or not MAIL(m) is free. Then, processing from step S207 to step S209 is repeated, as "m" is incremented till MAIL(m) with a free area is found. At step S209, when "m" has exceeded the maximum number, pointer "m" is set to the number of a mail with the earliest transmission date/time (step S210).

When MAIL(m) is a free area at step S207, or after "m" is set to a mail number at step S210, data of the mail creation area is stored in the area of MAIL(m) (step S211). Then, a transmission date/time of MAIL(m) is changed to the present date/time (step S212). Then, the display of the postscript mail creation screen image of FIG. 27 is cleared (step S213), the mail creation area is cleared (step S214) and this postscript mail creation processing is terminated.

At step S204, when a cursor position is not present in a position of the transmission switch, it is determined whether or not a cursor position is present

in a position of the cancellation switch (step S215). When the cursor position is present in this position, the display of the postscript mail creation screen image of FIG. 27 is cleared (step S213), the mail creation area is cleared (step S214) and this postscript mail creation processing is terminated. At step S215, when a cursor position is not present in a position of the cancellation switch either, program flow returns to step S194 of FIG. 25 and the mail creation screen image is displayed.

FIG. 28 is a flow for interrupt processing.

In this processing, it is determined whether or not a mail has been received in response to interrupts at given intervals (step S216) and when a mail has been received, it is determined whether or not a postscript flag of the received mail is "1" (step S217). When this flag is "1", pointer "n" is set to "1" (step S218) and it is determined whether or not a transmitter address of MAIL(n) of the received mail area is same as a transmitter address of a received MAIL (step S219).

When the transmitter address of MAIL(n) and the transmitter address of the received mail is same as each other, it is determined whether or not a subject of MAIL(n) and a subject of the received mail is same as each other (step S220). When both subjects are same, it is determined whether or not a unseal flag KAIFUF of MAIL(n) is "0" and has not been unsealed (step S221).

When both transmitter addresses are different from each other at step S219, when both subjects are different from each other at step S220 and when MAIL(n) has already been unsealed at step S221, "n" is incremented (step S222). Then, it is determined whether or not "n" has exceeded the maximum number (step S223) and when "n" is equal to or less than the maximum number, program flow goes to step S219 and only when a transmitter address and subject of MAIL(n) are same as those of a received mail and MAIL(n) has not been unsealed, a text of the received mail is added to (or integrated with) a text area of MAIL(n) for storage (step S224). Then, the received mail is cleared (step S225) and this interrupt processing is terminated. That is, when a received mail is a specific mail having a predetermined relation with MAIL(n), which has previously been received and stored in the received mail area, in other words when a received mail is a postscript mail of a previously received mail, texts of the received MAIL (a specific mail) and MAIL(n) are stored in an integrated manner.

At step S223, when "n" has exceeded the maximum number, that is, when MAIL(n), which has the same transmitter address and subject as a received mail, and which has not been unsealed, is not present or at step S217 when a postscript flag of a received mail is "0" and the received mail is not a specific mail, pointer

"n" is set to "1" (step S226) and an area of MAIL(n) of the received mail area is designated, as "n" is incremented. Then, it is determined whether or not the area of MAIL(n) is free (step S227).

When this area is not free, "n" is incremented (step S228) and it is determined whether or not "n" has exceeded the maximum number (step S229). When "n" is equal to or less than the maximum number, program flow returns to step S227 and it is determined whether or not an area of MAIL(n) is free. Then, a loop from step S227 to step S229 is repeated till a free area is found. When there is available no free area in the received mail area and "n" has exceeded the maximum number at step S229, an area number of a received mail with the earliest reception data/time is set to "n" (step S230).

At step S227 when an area of MAIL(n) is free, or at step S230 when an area number of a received mail is set to "n", received mail data other than a postscript of MAIL(n) is stored (step S231). Then, an unseal flag of MAIL(n) is reset to "0" (step S232) and this interrupt processing is terminated.

FIG. 29 is an unseal screen image of an integrated mail in which a text of a postscript mail is integrated with a text of a mail having a predetermined relation with the postscript mail and the integrated text is stored. This screen image is an unseal screen image in which the text on the postscript mail creation screen

image shown in FIG. 27 is integrated with the text on the detailed display screen image of the received mail shown in FIG. 14. FIG. 30 is a screen image of a postscript mail stored single in the received mail area when a mail having a specific relation with the postscript mail has already been unsealed.

As described above, according to the present embodiments, when a mail to be transmitted has a specific relation with a mail which has previously transmitted, a specific mail with a specific condition is created and transmitted without any description of a content of the previous mail. When a received mail is a specific mail which meets the specific condition and a mail with a predetermined relation with the specific mail has not been unsealed, the specific mail is integrated with the unsealed mail. Therefore, when a mail that has an addition to a previously transmitted mail is transmitted later, works in mail creation on the transmitter and in unsealing on the receiver can be made simple.

While in the above-mentioned embodiment, there is described a dedicated electronic mail apparatus, in ROM 2 of which there is stored a program for transmission/ reception of an electronic mail, a constitution can be adopted instead in which a program for transmission/ reception of an electronic mail is recorded on a floppy disk, a magneto-optical disk or the like and a personal

computer, a word processor or the like for a general purpose is operated on the program.

Further, while in the above-mentioned embodiment, when it is defined that a mail to be created is a postscript mail, a cursor position is made to be present in a position of the postscript switch icon on the screen image and then the mouse 6 is clicked, a constitution can be adopted instead in which a character string consisting of a single word "Postscript: " is inserted in a text of an ordinary mail creation processing and a mail with the symbol "Postscript: " in the text is automatically processed as a postscript mail. In this case, a mail creation work on the transmitter can be made simpler.

Industrial Applicability

According to the present invention, when a mail to be transmitted has a specific relation with a mail which has previously transmitted, a specific mail with a specific condition is created and transmitted without any description of a content of the previous mail. When a received mail is a specific mail which meets the specific condition and a mail with a predetermined relation with the specific mail has not been unsealed, the specific mail is integrated with the unsealed mail. Therefore, when a mail that has an addition to a previously transmitted mail is transmitted later, works in mail creation on the transmitter and in unsealing on

the receiver can be made simple.

C L A I M S

1. An electronic mail apparatus comprising:

ordinary mail creating means for creating an ordinary mail including a receiver address, a subject and a text;

ordinary mail transmitting means for transmitting the ordinary mail created by said ordinary mail creating means;

transmitted mail storage means for storing an ordinary mail transmitted by said ordinary mail transmitting means;

transmitted mail designating means for designating one of transmitted mails stored in said transmitted mail storage means;

postscript mail creating means for creating a postscript mail which is same as the mail designated by said transmitted mail designating means with respect to at least one of a receiver address and subject of the designated mail;

postscript mail transmitting means for transmitting the postscript mail created by said postscript mail creating means and storing a transmitted postscript mail in the transmitted mail storage means;

mail receiving means for receiving a mail;

received mail storage means for storing the mail received by said mail receiving means;

received mail designating means for designating one of received mails stored in said received mail storage means; and

unsealing means for displaying at least one of a transmitter address, subject and text of the received mail designated by said received mail designating means.

2. An electronic mail apparatus according to claim 1, wherein said postscript mail creating means comprises copying means for copying at least one of a receiver address and subject of the transmitted mail designated by said transmitted mail designating means on a postscript mail to be created.

3. An electronic mail apparatus according to claim 1, wherein said postscript mail creating means comprises automatic character writing means for writing characters indicating that a mail is a postscript mail in a text of the mail.

4. An electronic mail apparatus comprising:

mail creating means for creating an ordinary mail including a receiver address, a subject and a text and a postscript mail having a predetermined relation with a previously transmitted mail;

mail transmitting means for transmitting the mail created by said mail creating means;

mail receiving means for receiving a mail;

mail storage means for storing the mail received by said mail receiving means;

unsealing means for designating one of the received mails stored in said mail storage means and displaying at least one of a transmitter address, subject and text of the designated mail;

specific mail detecting means for detecting whether or not the mail received by said mail receiving means is a postscript mail;

mail extracting means for extracting a mail which has a predetermined relation with the postscript mail among the received mails stored in said mail storage means when the mail detected by said postscript mail detecting means is the postscript mail;

unseal detecting means for detecting whether or not the mail extracted by said mail extracting means has been unsealed by said unsealing means; and

storage control means for integrating the extracted mail and the received mail with relating to each other in the mail storage means when the extracted mail is detected as having not been unsealed by said unseal detecting means, and storing the received mail in said mail storage means when the mail detected by said postscript mail detecting means is not the postscript mail or when the extracted mail is detected as having been unsealed by said unseal detecting means.

5. An electronic mail apparatus according to claim 4, wherein said mail extracting means extracts a mail which is same as the postscript mail with respect

to at least one of a transmitter address and subject of the postscript mail.

6. An electronic mail apparatus according to claim 4, wherein said storage control means adds a text of the postscript mail to a text of the related mail when the postscript mail is stored in said mail storage means.

7. A storage medium readable by a computer storing a program for electronic mail processing, the program comprising:

a program code A for creating an ordinary mail including a receiver address, a subject and a text;

a program code B for transmitting the ordinary mail created by said program code A;

a program code C for storing the ordinary mail transmitted by said program code B;

a program code D for designating one of transmitted mails stored by said program code C;

a program code E for creating a postscript mail which is same as the mail designated by said program code D with respect to at least one of a receiver address and subject of the designated mail;

a program code F for transmitting the postscript mail created by said program code E and storing a transmitted postscript mail;

a program code G for receiving a mail;

a program code H for storing a mail received by

said program code G;

a program code I for designating one of received mails stored by said program code H; and

a program code J for displaying at least one of a transmitter address, subject and text of the received mail designated by said program code I.

8. A storage medium readable by a computer storing a program for electronic mail processing, the program comprising:

a program code A for creating an ordinary mail including a receiver address, a subject and a text but also a postscript mail having a predetermined relation with a previously transmitted mail;

a program code B for transmitting the mail created by said program code A;

a program code C for receiving a mail;

a program code D for storing the mail received by said program code C;

a program code E for designating one of the received mails stored by said program code D and displaying at least one of a transmitter address, subject and text of the designated mail;

a program code F for detecting whether or not the mail received by said program code C is a postscript mail;

a program code G for extracting a mail which has a predetermined relation with the postscript mail among

the received mails stored by said program code D when the mail detected by said program code F is the postscript mail;

a program code H for detecting whether or not the mail extracted by said program code G has been unsealed; and

a program code I for integrating the extracted mail and the received mail with relating to each other when the extracted mail is detected as having not been unsealed by said program code H, and storing the received mail when the mail detected by said program code F is not the specific mail or when the extracted mail is detected as having been unsealed by said program code H.

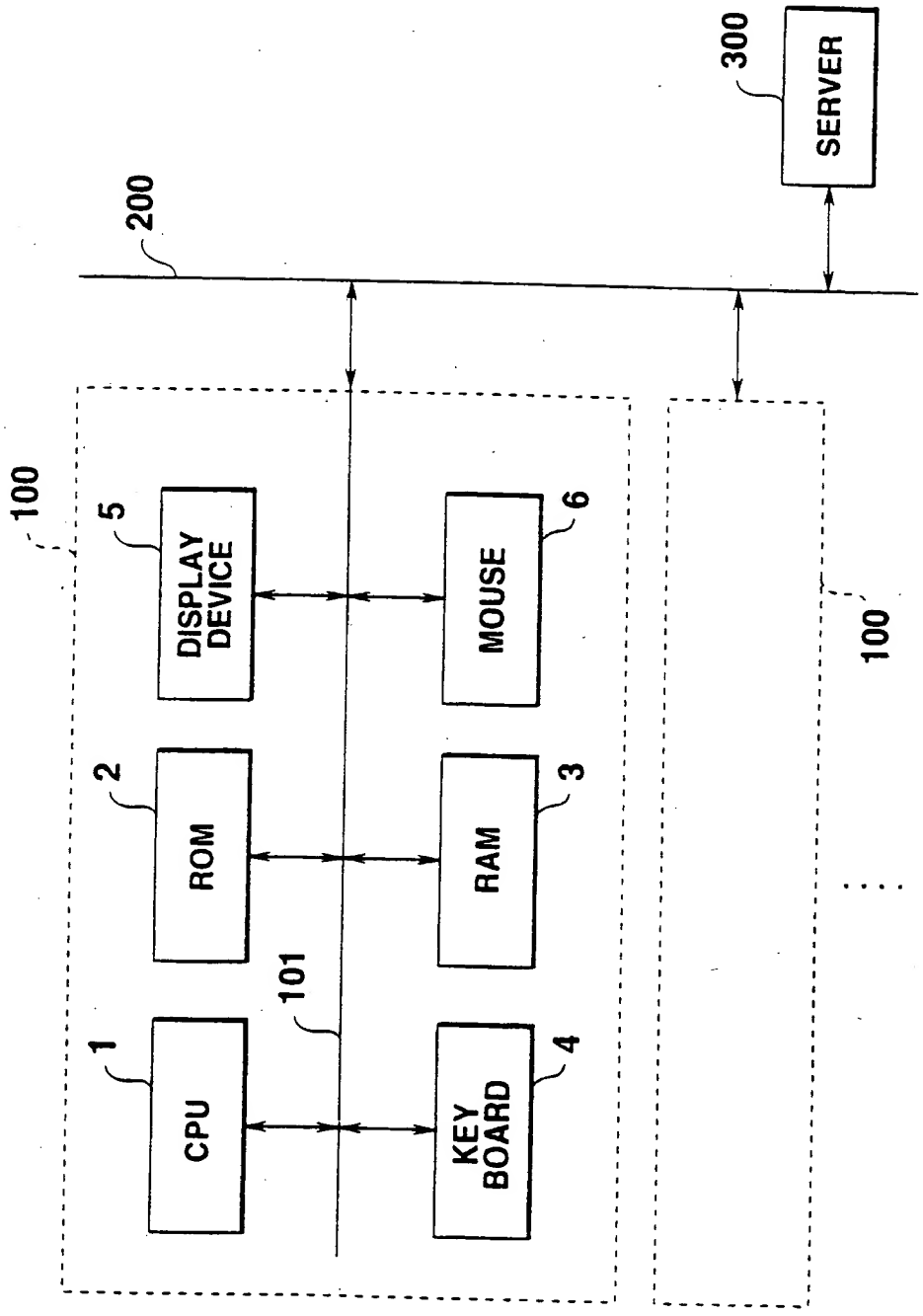

300
SERVER
200
100
5
DISPLAY DEVICE
6
MOUSE
2
ROM
3
RAM
101
1
CPU
4
KEY BOARD
100

FIG.1

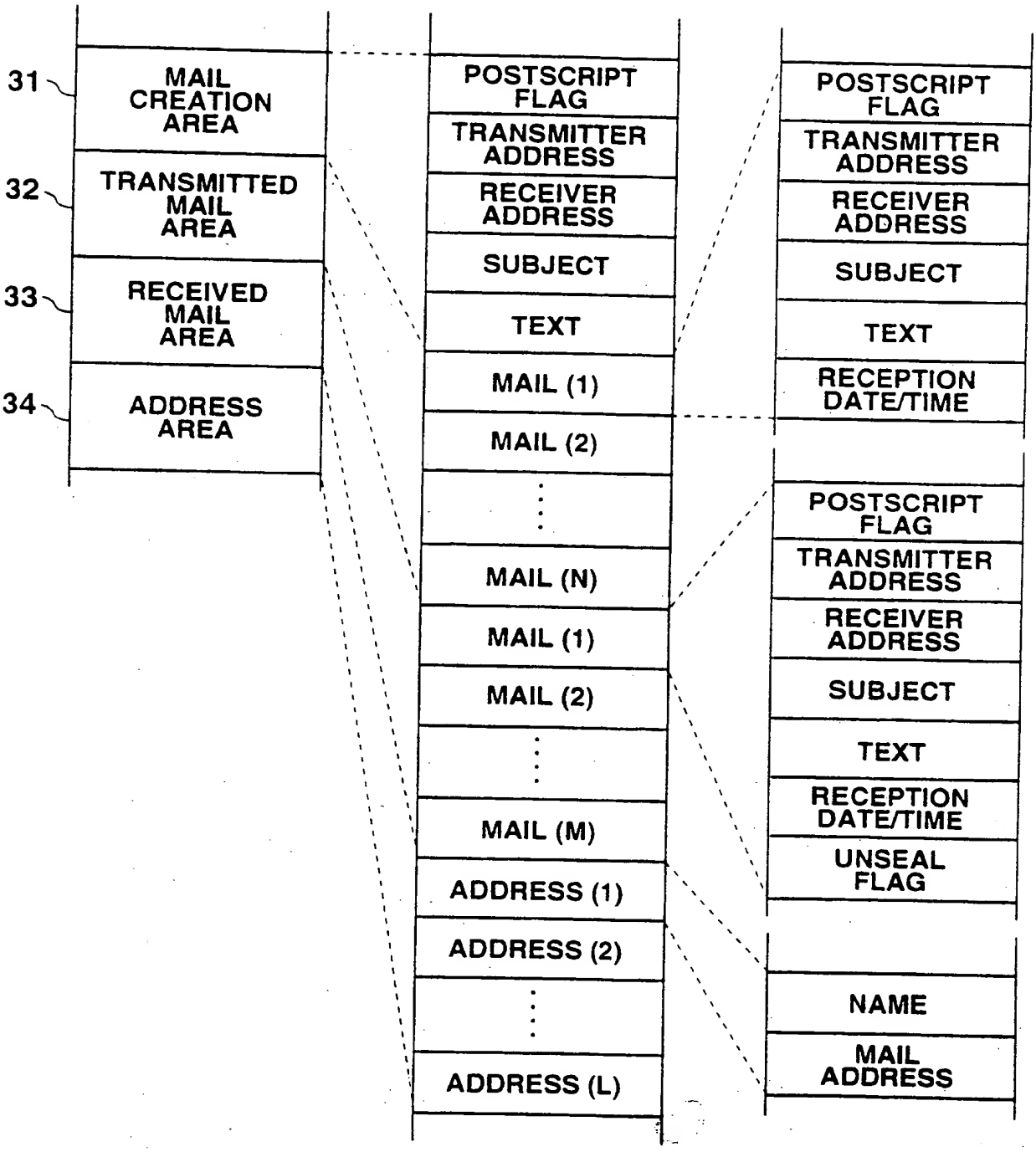


FIG.2

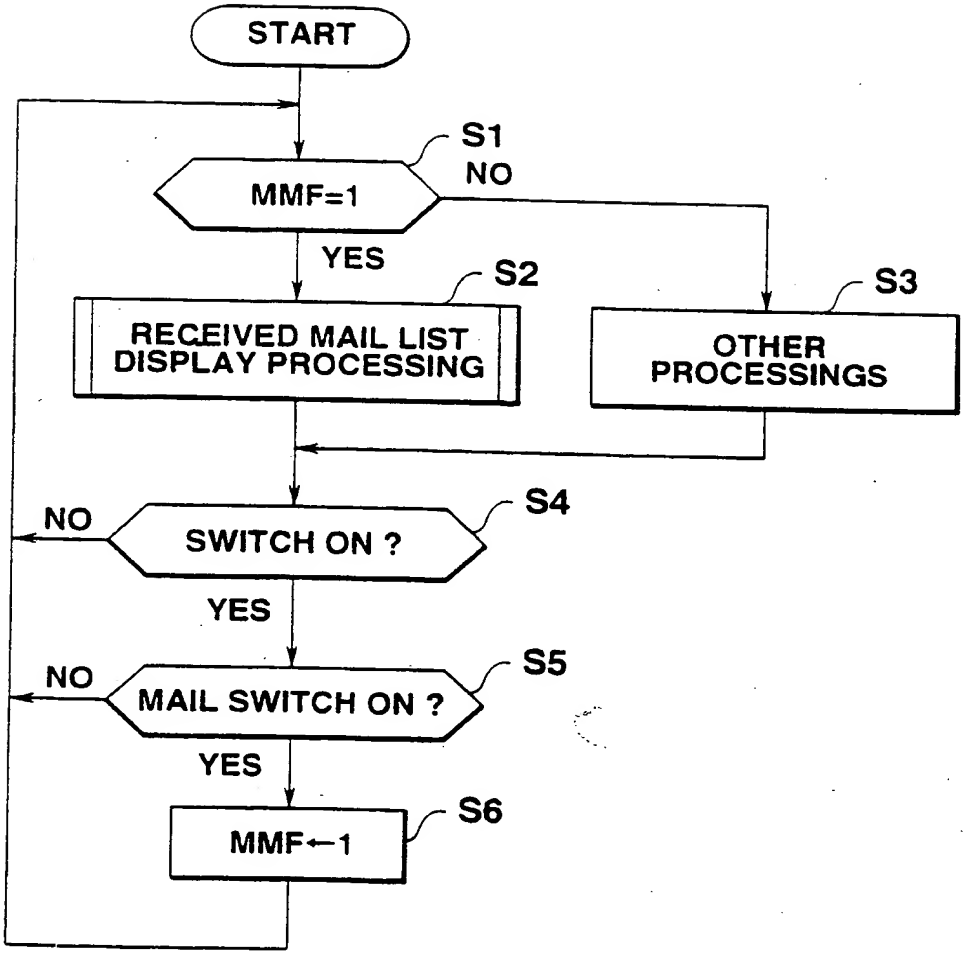

START
S1
MMF=1
NO
YES
S2
RECEIVED MAIL LIST DISPLAY PROCESSING
S3
OTHER PROCESSINGS
NO
SWITCH ON ?
S4
YES
NO
MAIL SWITCH ON ?
S5
YES
MMF←1
S6

FIG.3

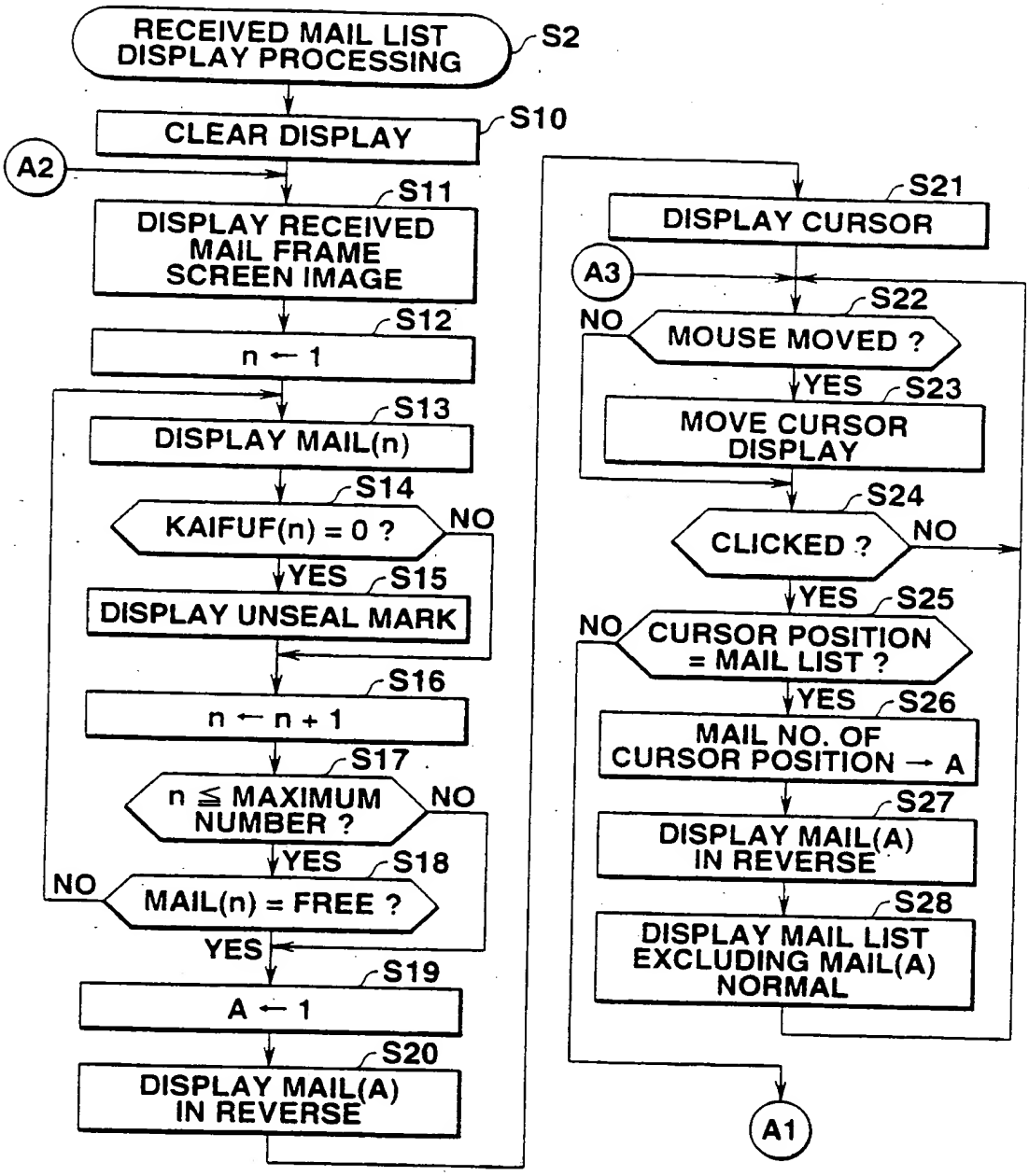


FIG.4

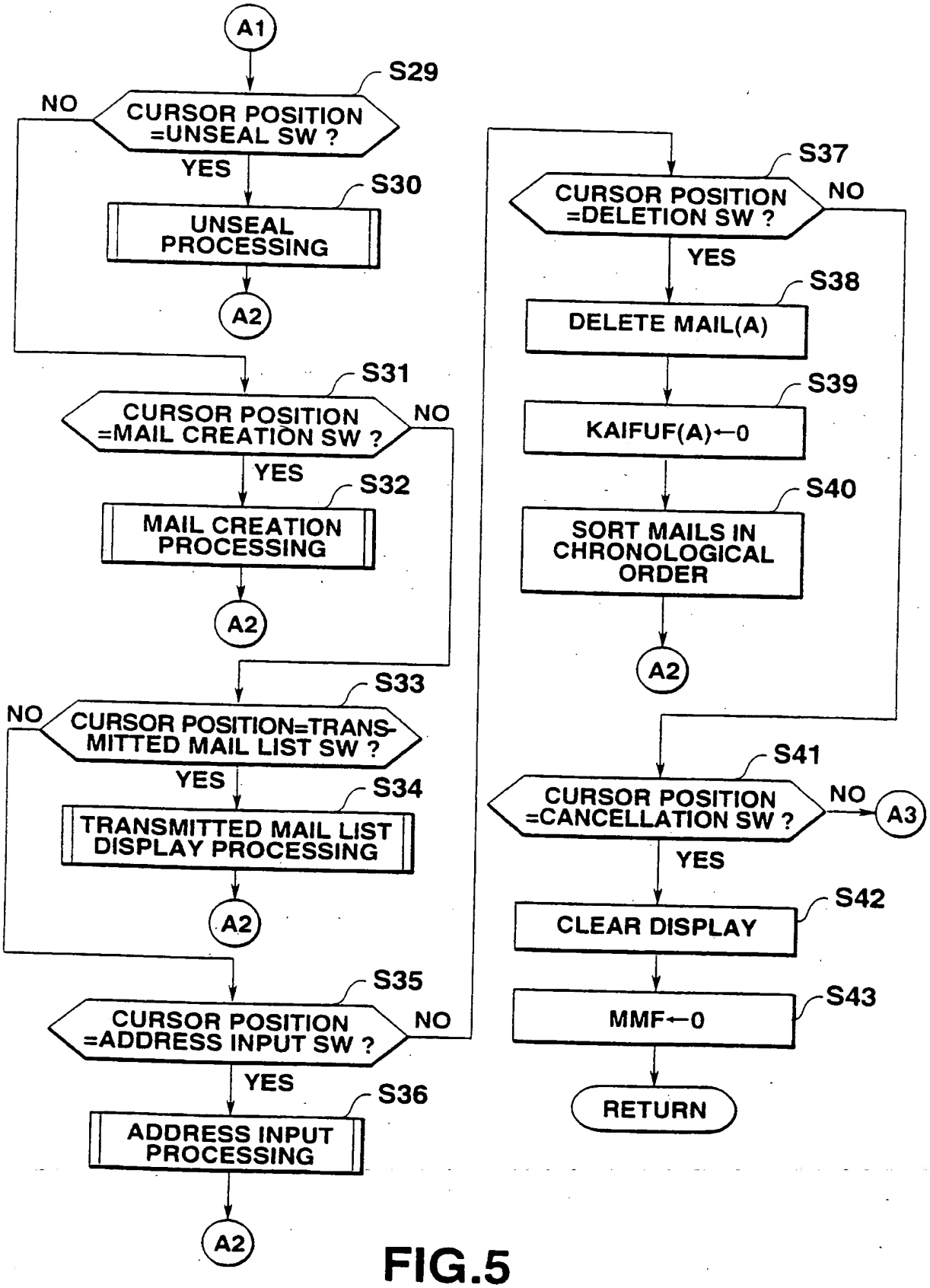

A1
S29
NO
CURSOR POSITION =UNSEAL SW ?
YES
S30
UNSEAL PROCESSING
A2
S31
CURSOR POSITION =MAIL CREATION SW ?
NO
YES
S32
MAIL CREATION PROCESSING
A2
S33
NO
CURSOR POSITION=TRANSMITTED MAIL LIST SW ?
YES
S34
TRANSMITTED MAIL LIST DISPLAY PROCESSING
A2
S35
CURSOR POSITION =ADDRESS INPUT SW ?
NO
YES
S36
ADDRESS INPUT PROCESSING
A2
S37
CURSOR POSITION =DELETION SW ?
NO
YES
S38
DELETE MAIL(A)
S39
KAIFUF(A)←0
S40
SORT MAILS IN CHRONOLOGICAL ORDER
A2
S41
CURSOR POSITION =CANCELLATION SW ?
NO
A3
YES
S42
CLEAR DISPLAY
S43
MMF←0
RETURN

FIG.5

RECEIVED MAIL LIST SCREEN IMAGE

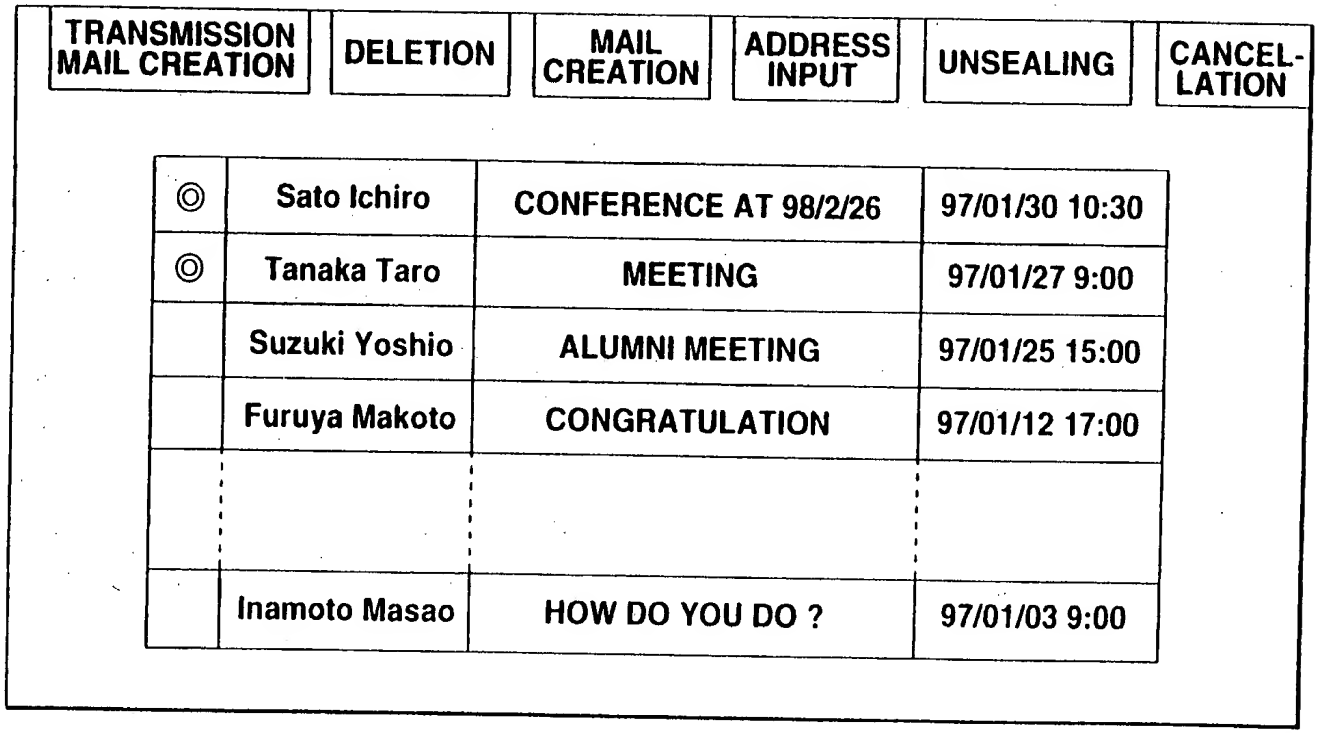


FIG.6

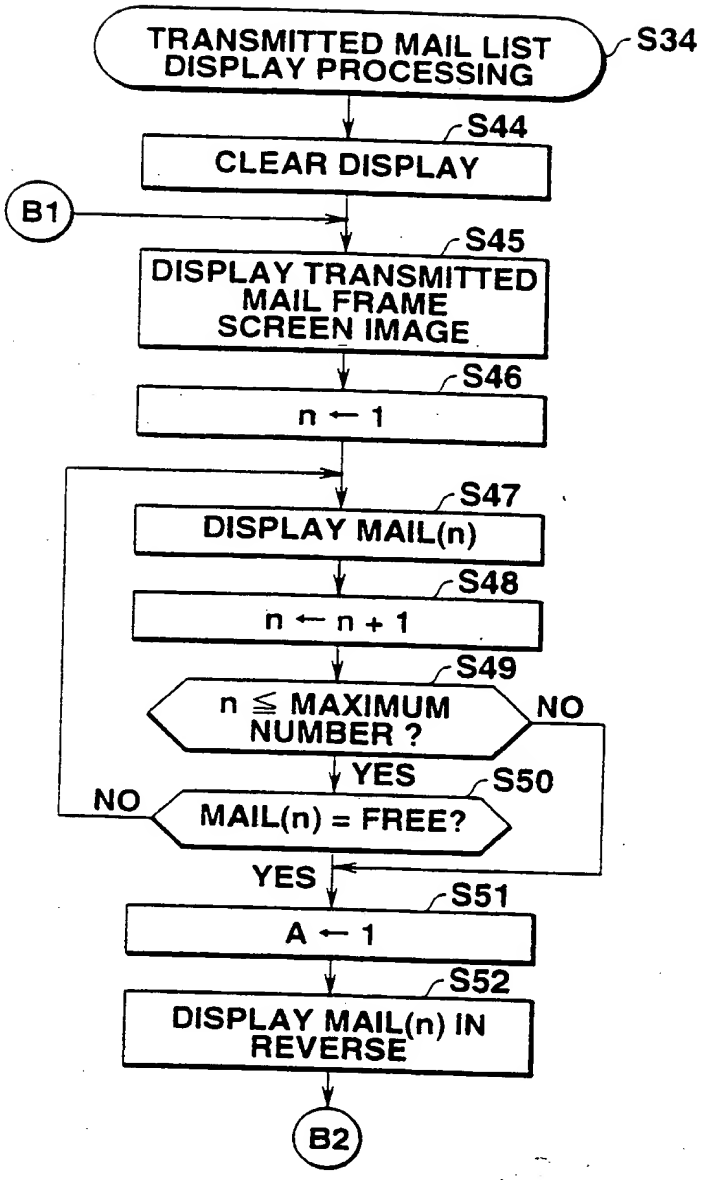

TRANSMITTED MAIL LIST DISPLAY PROCESSING
S34
S44
CLEAR DISPLAY
B1
S45
DISPLAY TRANSMITTED MAIL FRAME SCREEN IMAGE
S46
n ← 1
S47
DISPLAY MAIL(n)
S48
n ← n + 1
S49
n ≦ MAXIMUM NUMBER ?
NO
YES
S50
NO
MAIL(n) = FREE?
YES
S51
A ← 1
S52
DISPLAY MAIL(n) IN REVERSE
B2

FIG.7

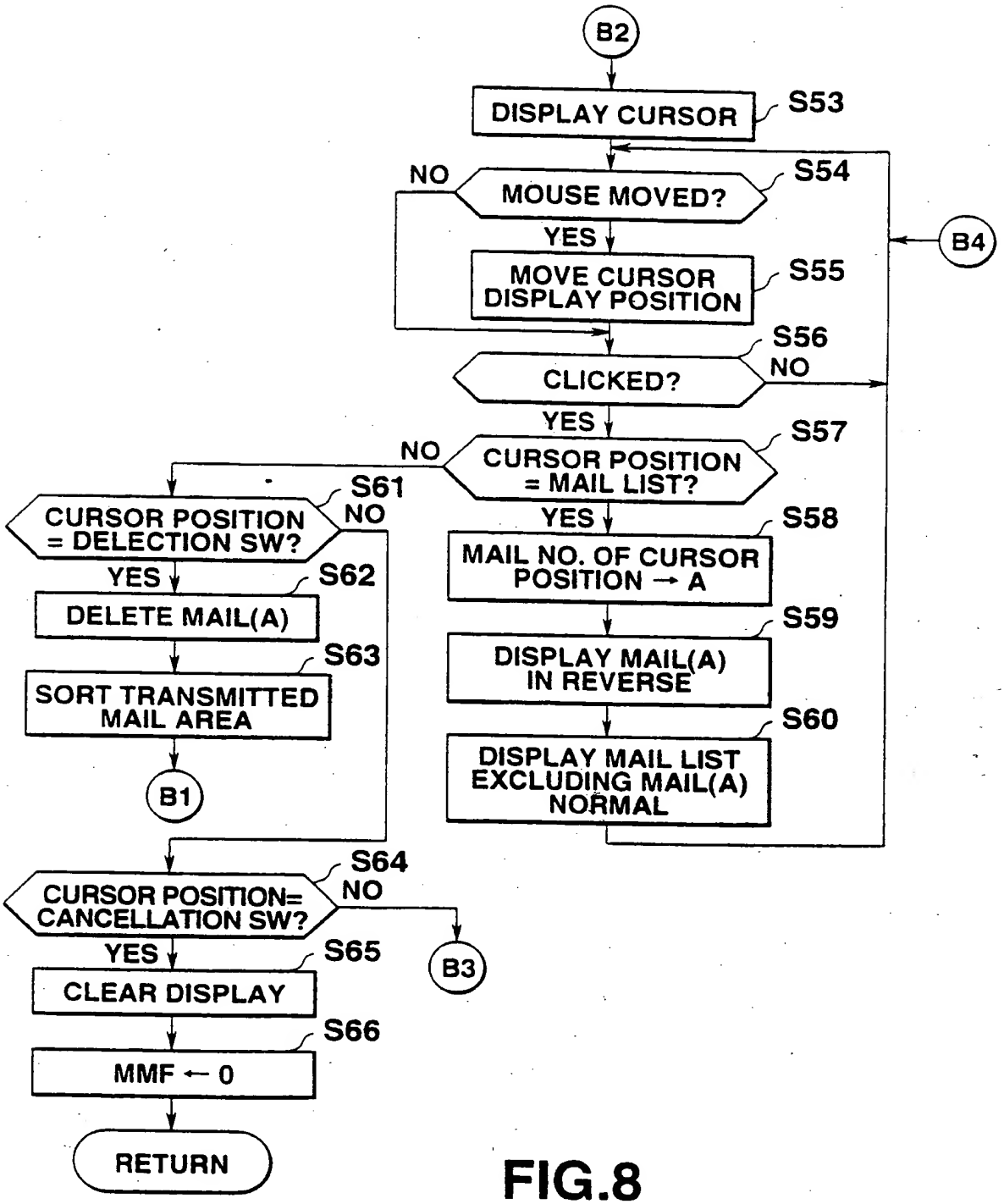


FIG.8

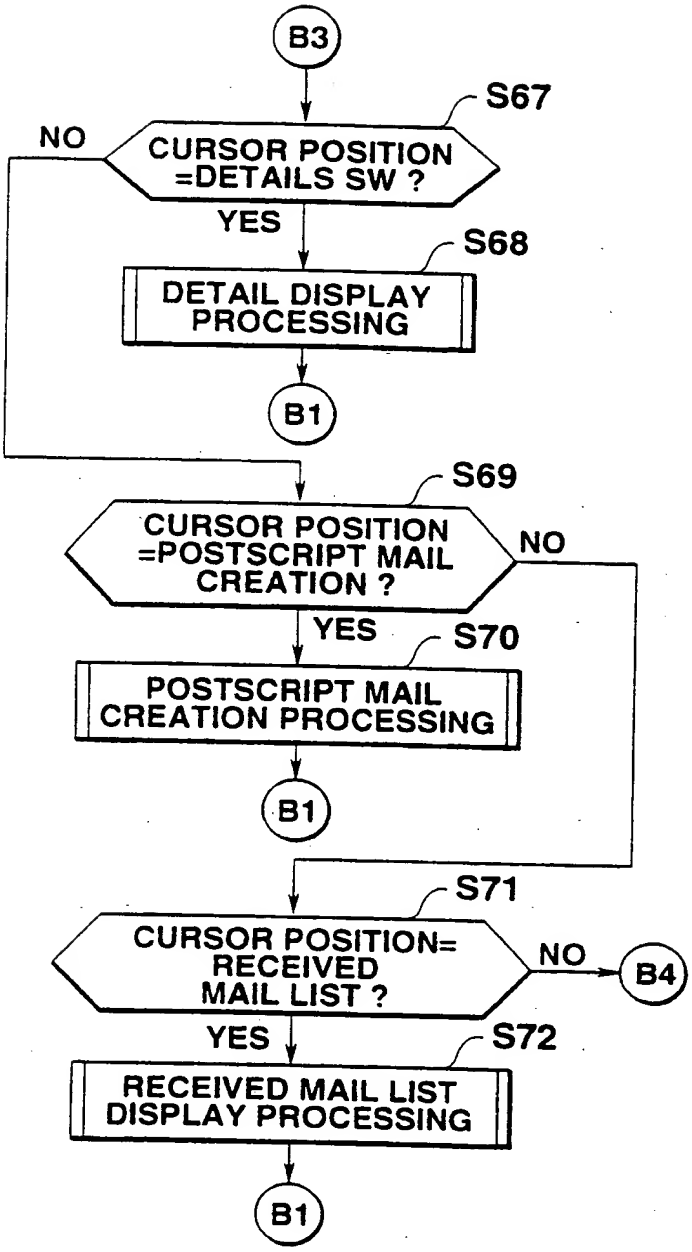

B3
S67
NO
CURSOR POSITION =DETAILS SW ?
YES
S68
DETAIL DISPLAY PROCESSING
B1
S69
CURSOR POSITION =POSTSCRIPT MAIL CREATION ?
NO
YES
S70
POSTSCRIPT MAIL CREATION PROCESSING
B1
S71
CURSOR POSITION= RECEIVED MAIL LIST ?
NO
B4
YES
S72
RECEIVED MAIL LIST DISPLAY PROCESSING
B1

FIG.9

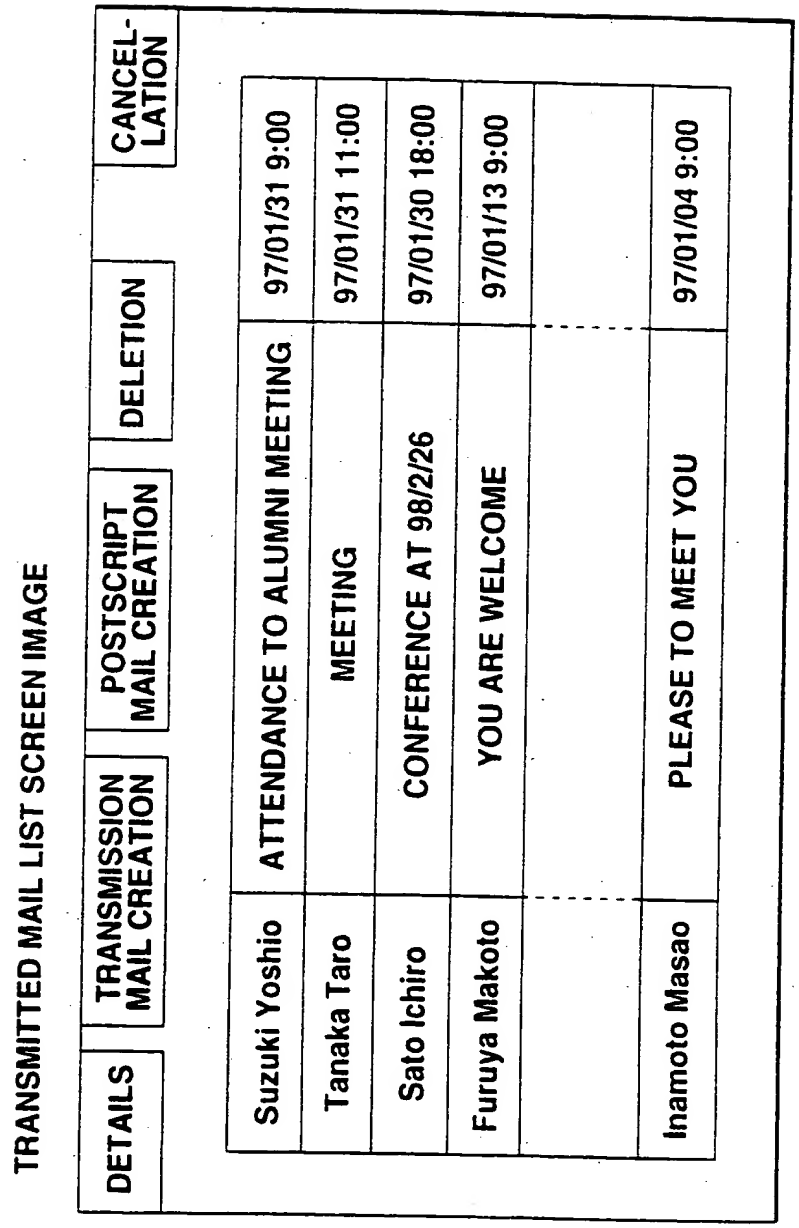


FIG.10

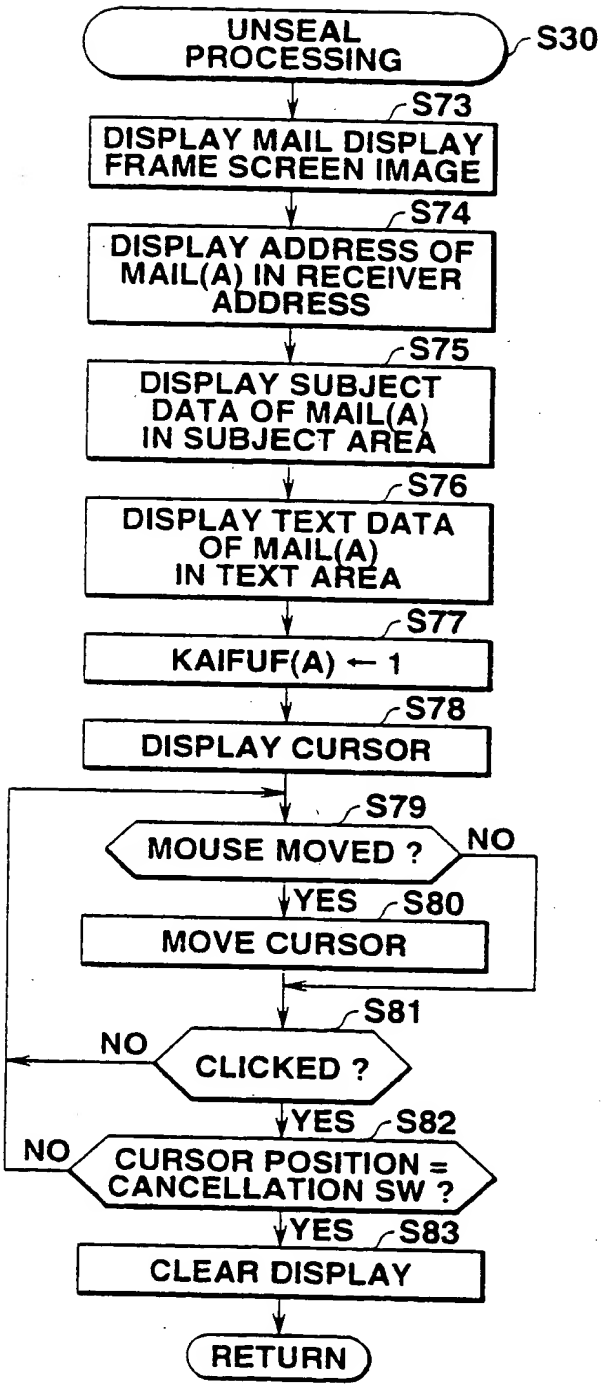

UNSEAL PROCESSING
S30
S73
DISPLAY MAIL DISPLAY FRAME SCREEN IMAGE
S74
DISPLAY ADDRESS OF MAIL(A) IN RECEIVER ADDRESS
S75
DISPLAY SUBJECT DATA OF MAIL(A) IN SUBJECT AREA
S76
DISPLAY TEXT DATA OF MAIL(A) IN TEXT AREA
S77
KAIFUF(A) ← 1
S78
DISPLAY CURSOR
S79
MOUSE MOVED ?
NO
YES
S80
MOVE CURSOR
S81
NO
CLICKED ?
YES
S82
NO
CURSOR POSITION = CANCELLATION SW ?
YES
S83
CLEAR DISPLAY
RETURN

FIG.11

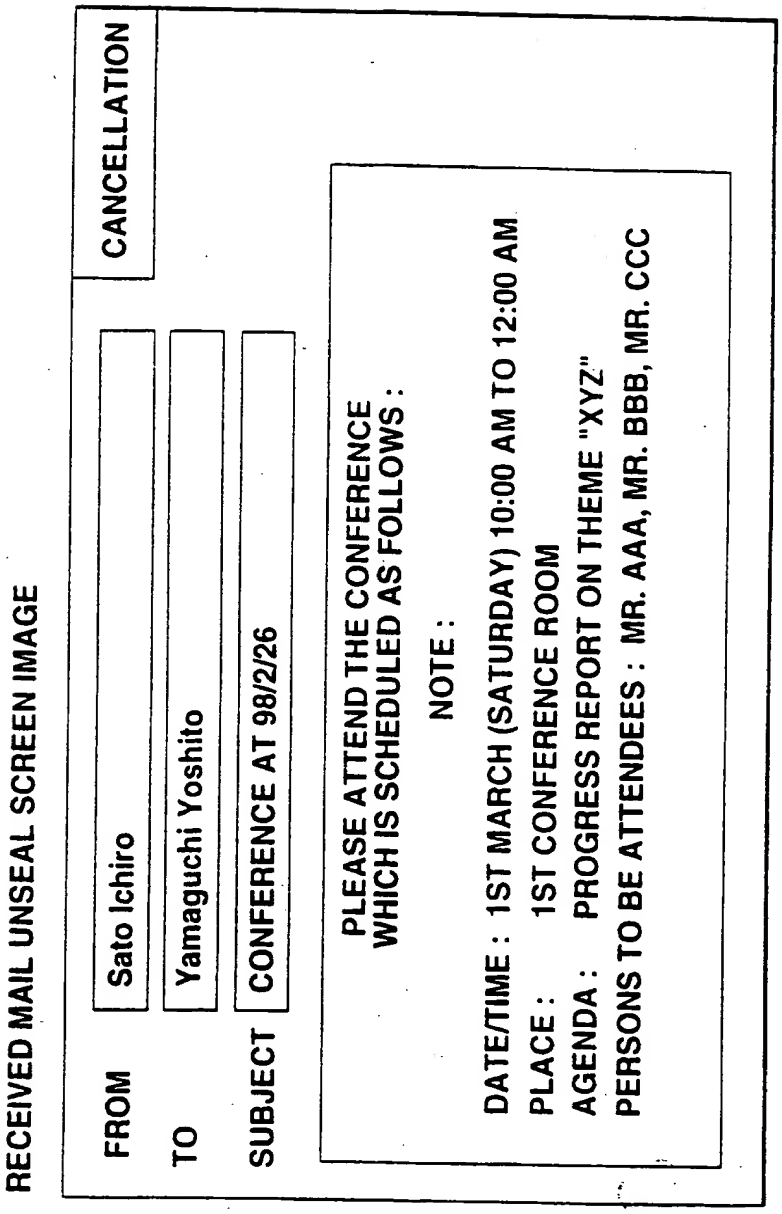


FIG.12

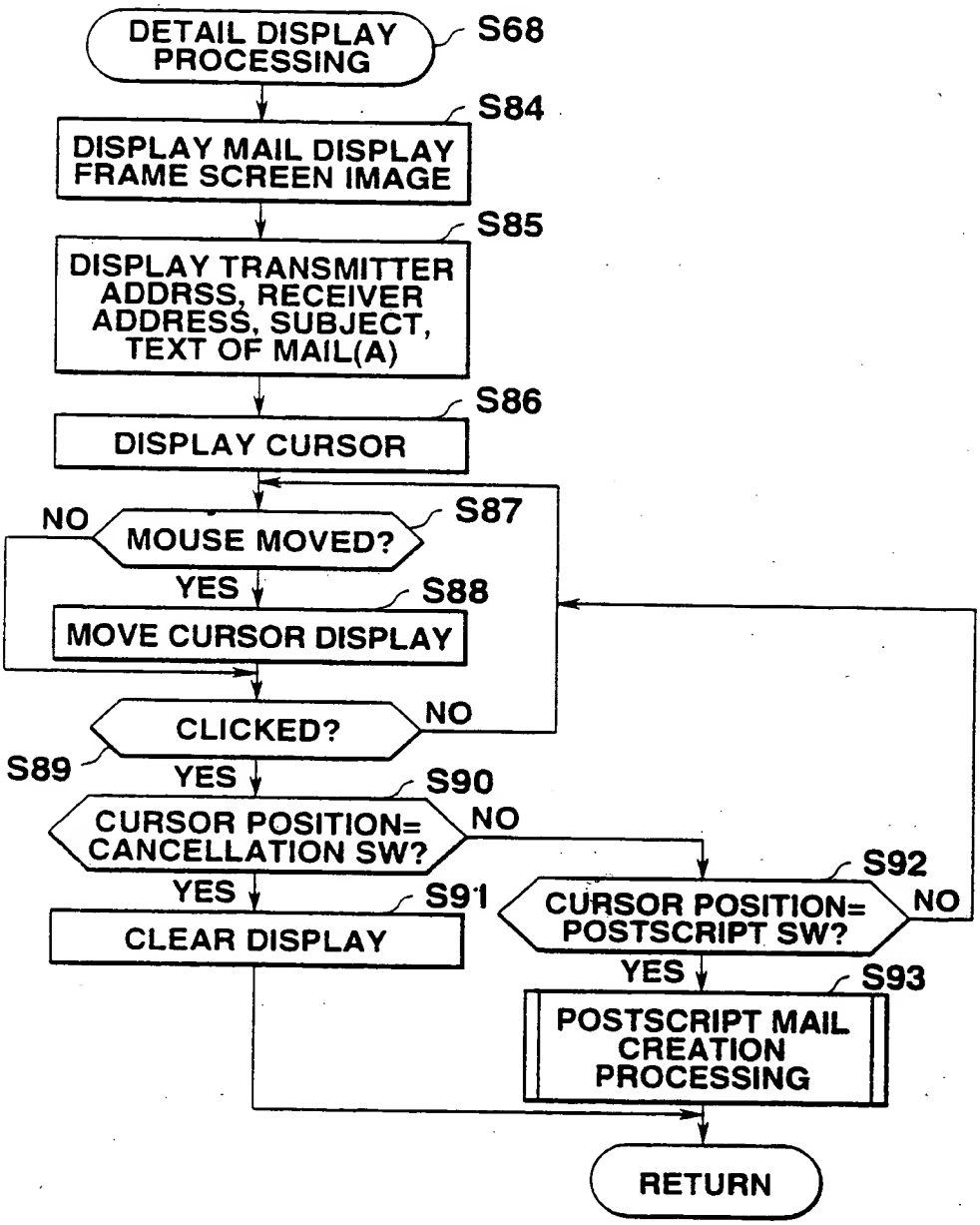


FIG.13

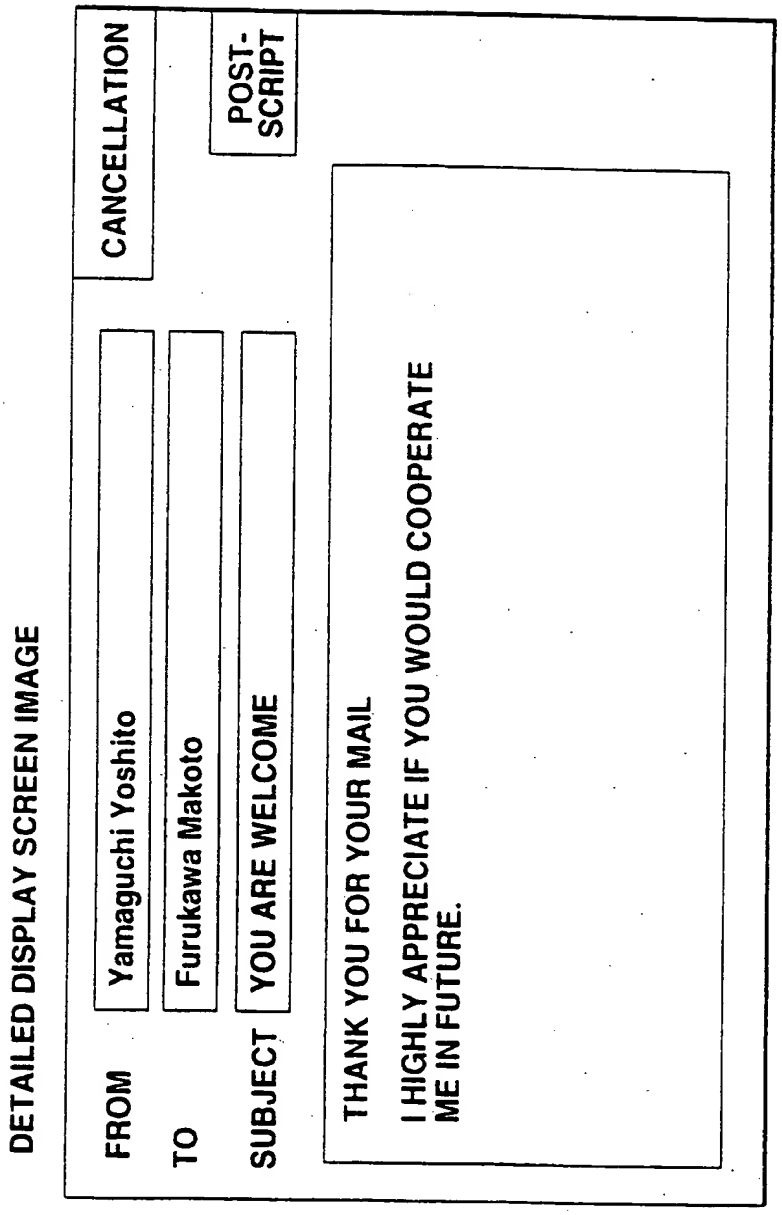

DETAILED DISPLAY SCREEN IMAGE
FROM
Yamaguchi Yoshito
TO
Furukawa Makoto
SUBJECT
YOU ARE WELCOME
CANCELLATION
POST-SCRIPT
THANK YOU FOR YOUR MAIL
I HIGHLY APPRECIATE IF YOU WOULD COOPERATE ME IN FUTURE.

FIG.14

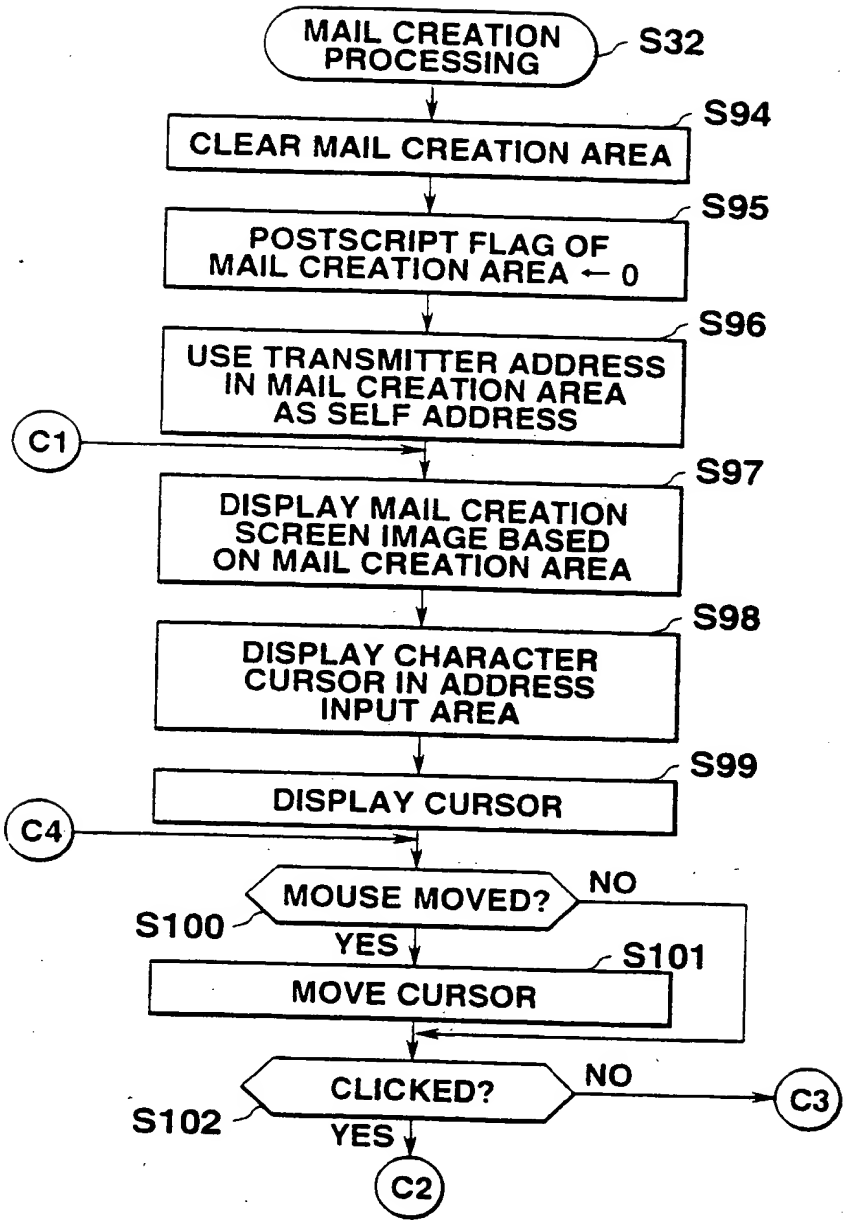

MAIL CREATION PROCESSING
S32
S94
CLEAR MAIL CREATION AREA
S95
POSTSCRIPT FLAG OF MAIL CREATION AREA ← 0
S96
USE TRANSMITTER ADDRESS IN MAIL CREATION AREA AS SELF ADDRESS
C1
S97
DISPLAY MAIL CREATION SCREEN IMAGE BASED ON MAIL CREATION AREA
S98
DISPLAY CHARACTER CURSOR IN ADDRESS INPUT AREA
S99
DISPLAY CURSOR
C4
MOUSE MOVED?
NO
S100
YES
S101
MOVE CURSOR
CLICKED?
NO
C3
S102
YES
C2

FIG.15

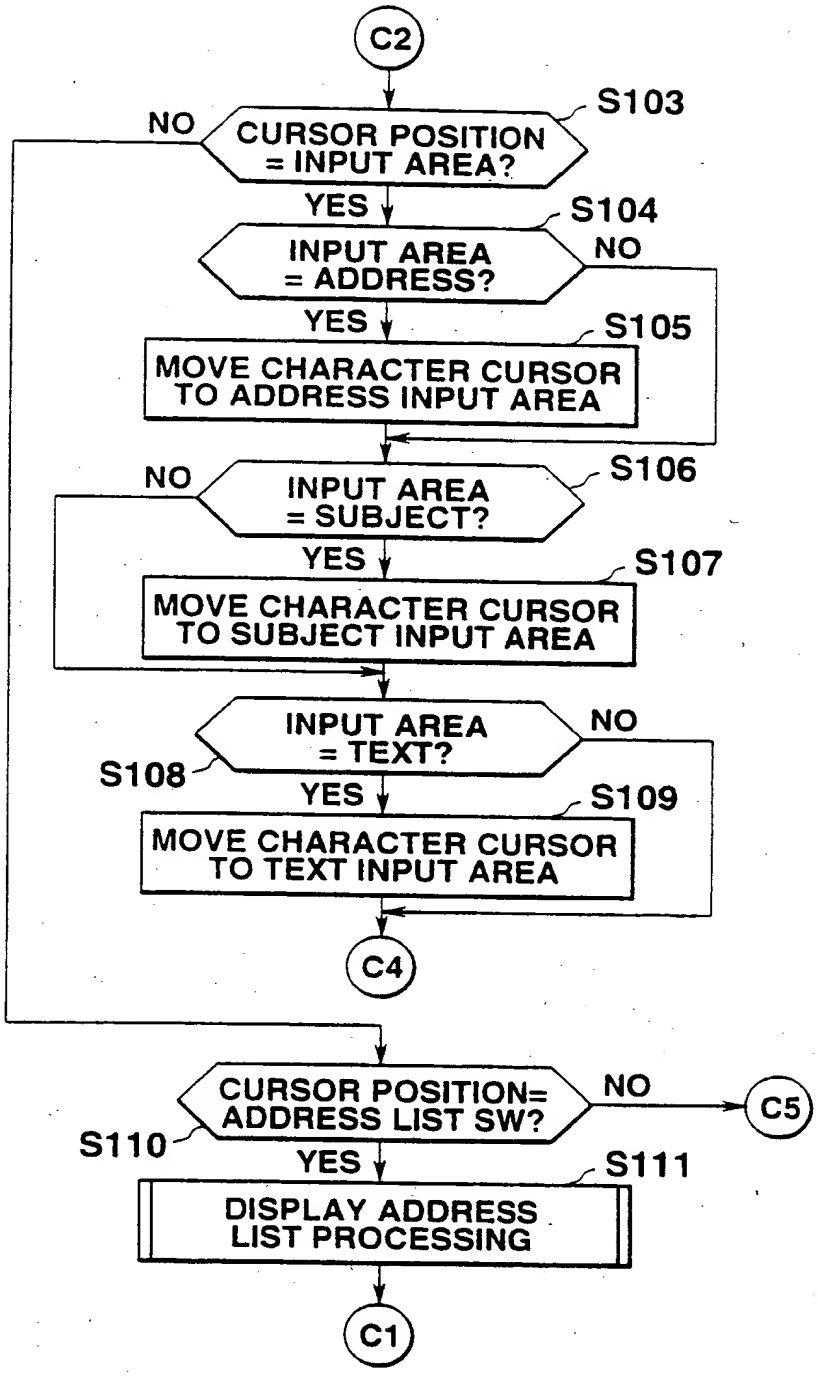


FIG.16

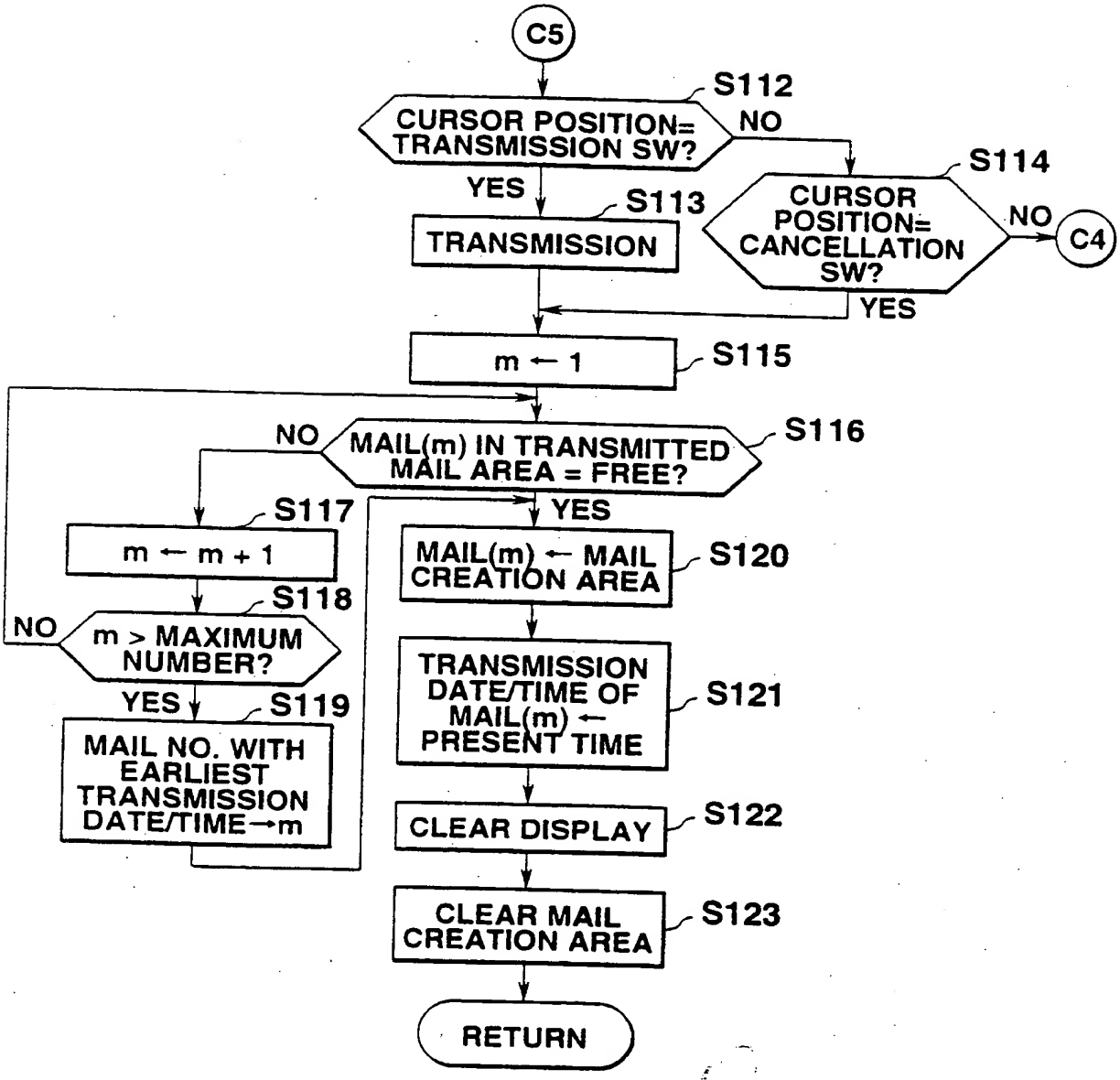

C5
S112
CURSOR POSITION= TRANSMISSION SW?
NO
S114
CURSOR POSITION= CANCELLATION SW?
NO
C4
YES
S113
TRANSMISSION
YES
m ← 1
S115
NO
MAIL(m) IN TRANSMITTED MAIL AREA = FREE?
S116
YES
S117
m ← m + 1
S118
NO
m > MAXIMUM NUMBER?
YES
S119
MAIL NO. WITH EARLIEST TRANSMISSION DATE/TIME→m
MAIL(m) ← MAIL CREATION AREA
S120
TRANSMISSION DATE/TIME OF MAIL(m) ← PRESENT TIME
S121
CLEAR DISPLAY
S122
CLEAR MAIL CREATION AREA
S123
RETURN

FIG.17

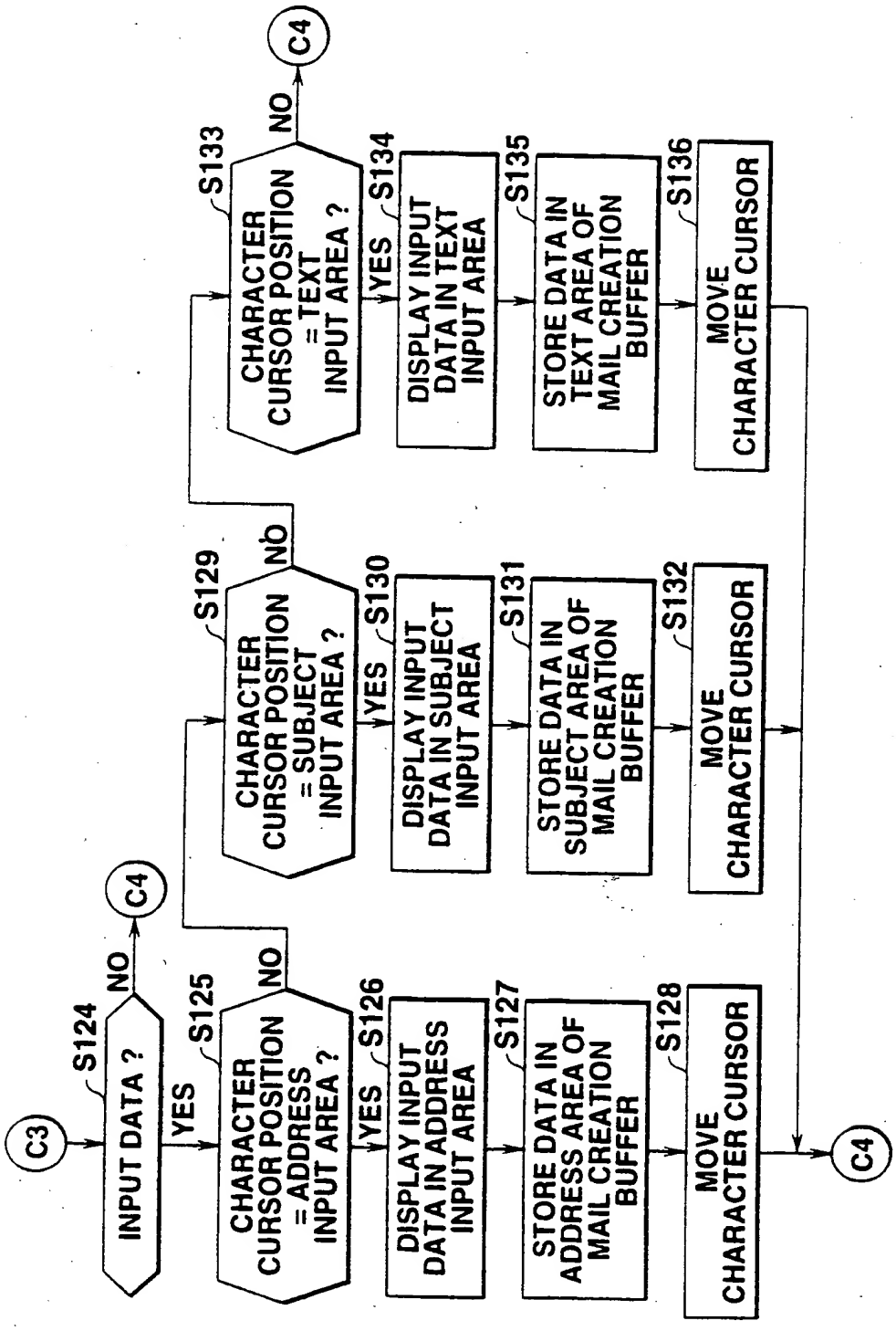


FIG.18

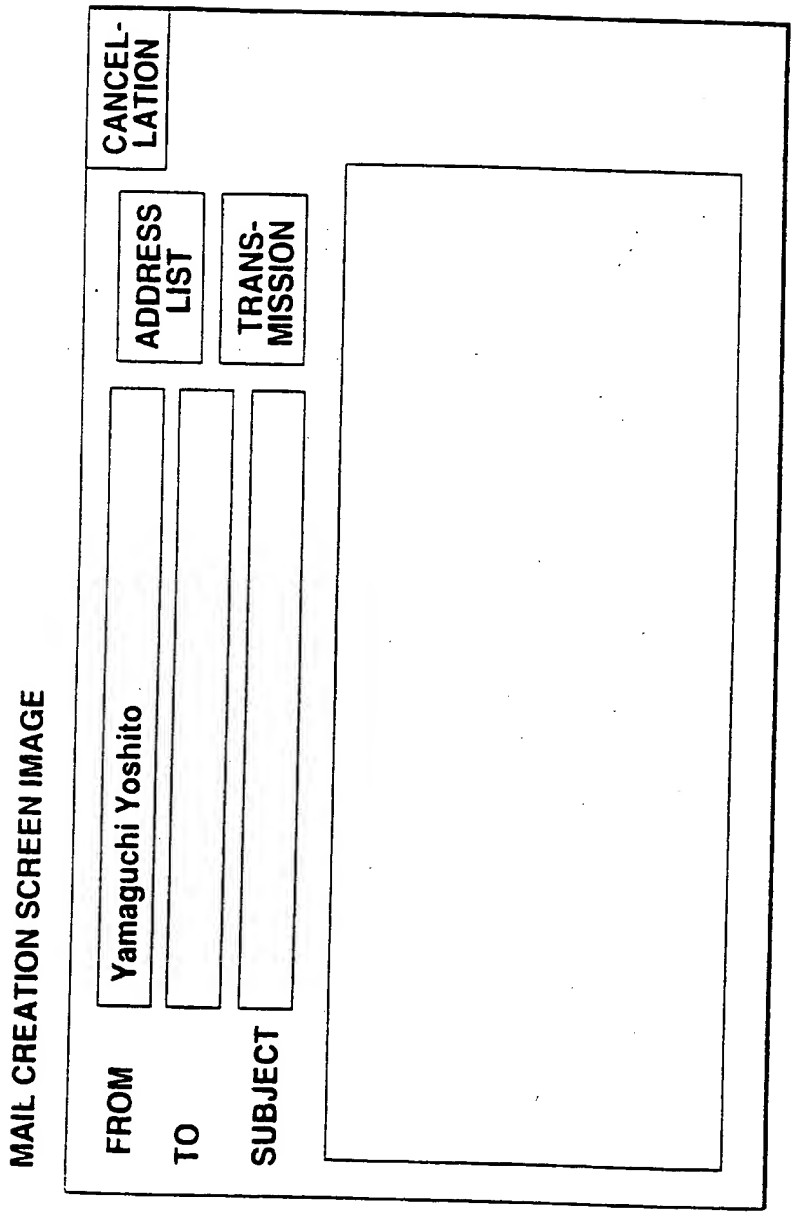

MAIL CREATION SCREEN IMAGE
FROM
Yamaguchi Yoshito
TO
SUBJECT
ADDRESS LIST
TRANS-MISSION
CANCEL-LATION

FIG.19

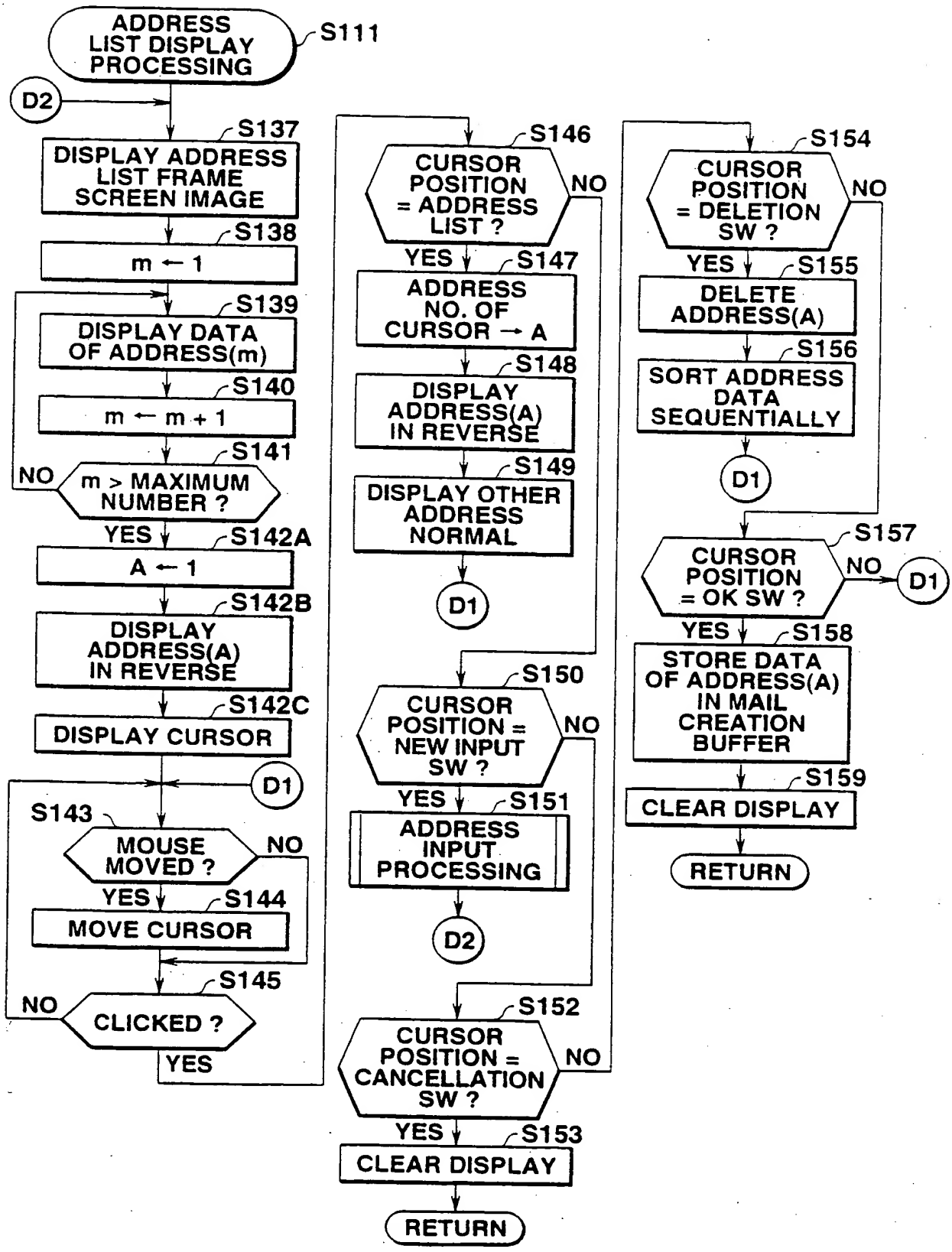


FIG.20

ADDRESS LIST DISPLAY SCREEN IMAGE

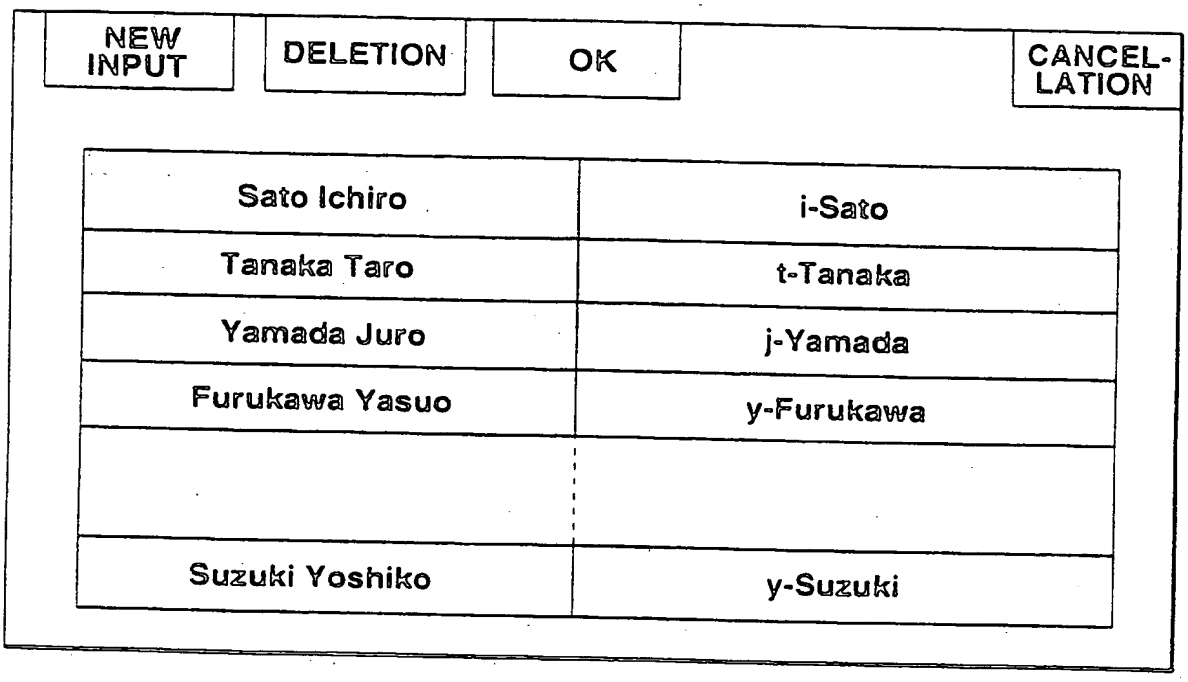


FIG.21

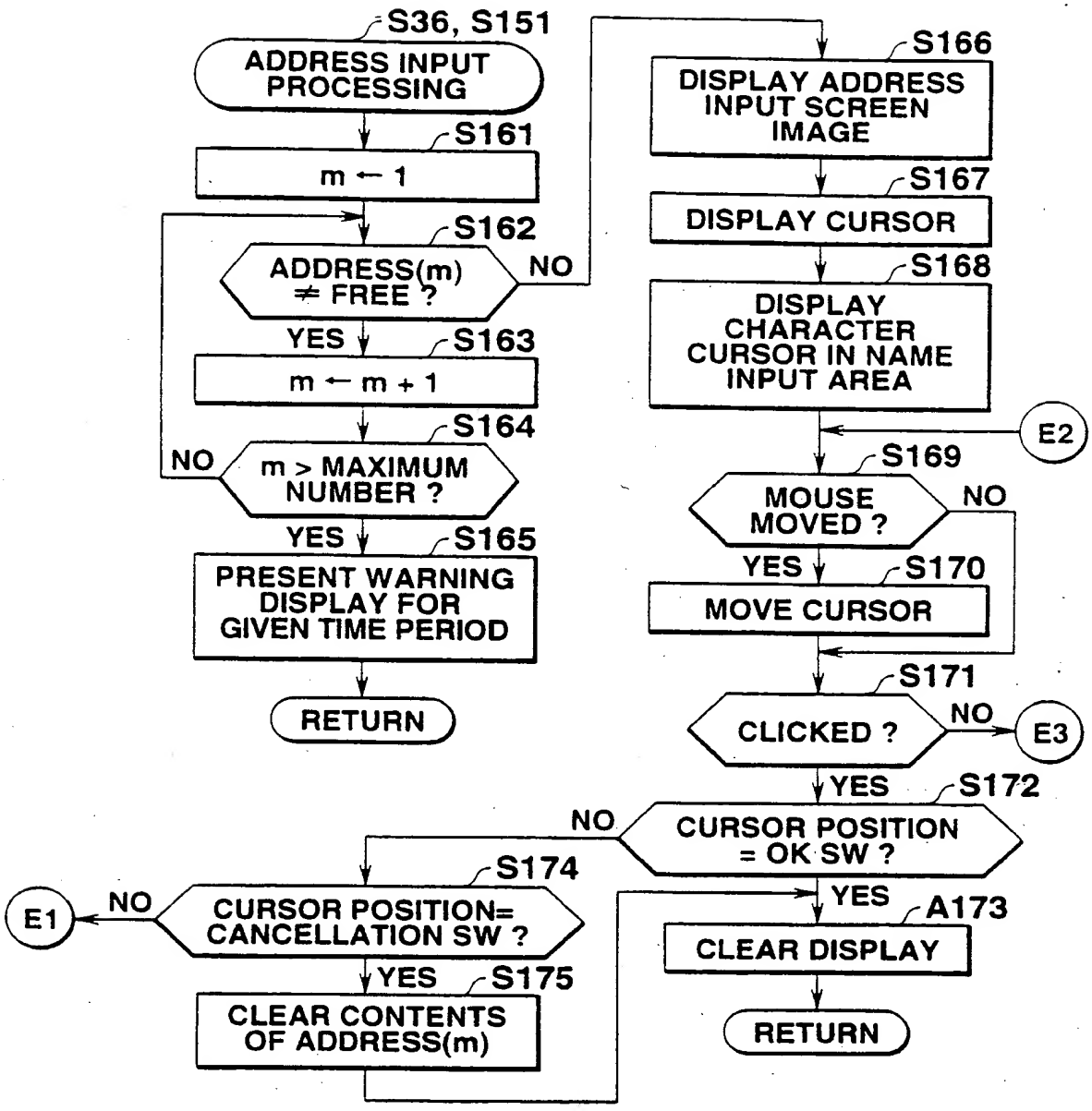

S36, S151
ADDRESS INPUT PROCESSING
S161
m ← 1
S162
ADDRESS(m) ≠ FREE ?
NO
YES
S163
m ← m + 1
S164
NO
m > MAXIMUM NUMBER ?
YES
S165
PRESENT WARNING DISPLAY FOR GIVEN TIME PERIOD
RETURN
S166
DISPLAY ADDRESS INPUT SCREEN IMAGE
S167
DISPLAY CURSOR
S168
DISPLAY CHARACTER CURSOR IN NAME INPUT AREA
E2
S169
MOUSE MOVED ?
NO
YES
S170
MOVE CURSOR
S171
CLICKED ?
NO
E3
YES
S172
NO
CURSOR POSITION = OK SW ?
S174
E1
NO
CURSOR POSITION= CANCELLATION SW ?
YES
S175
CLEAR CONTENTS OF ADDRESS(m)
YES
A173
CLEAR DISPLAY
RETURN

FIG.22

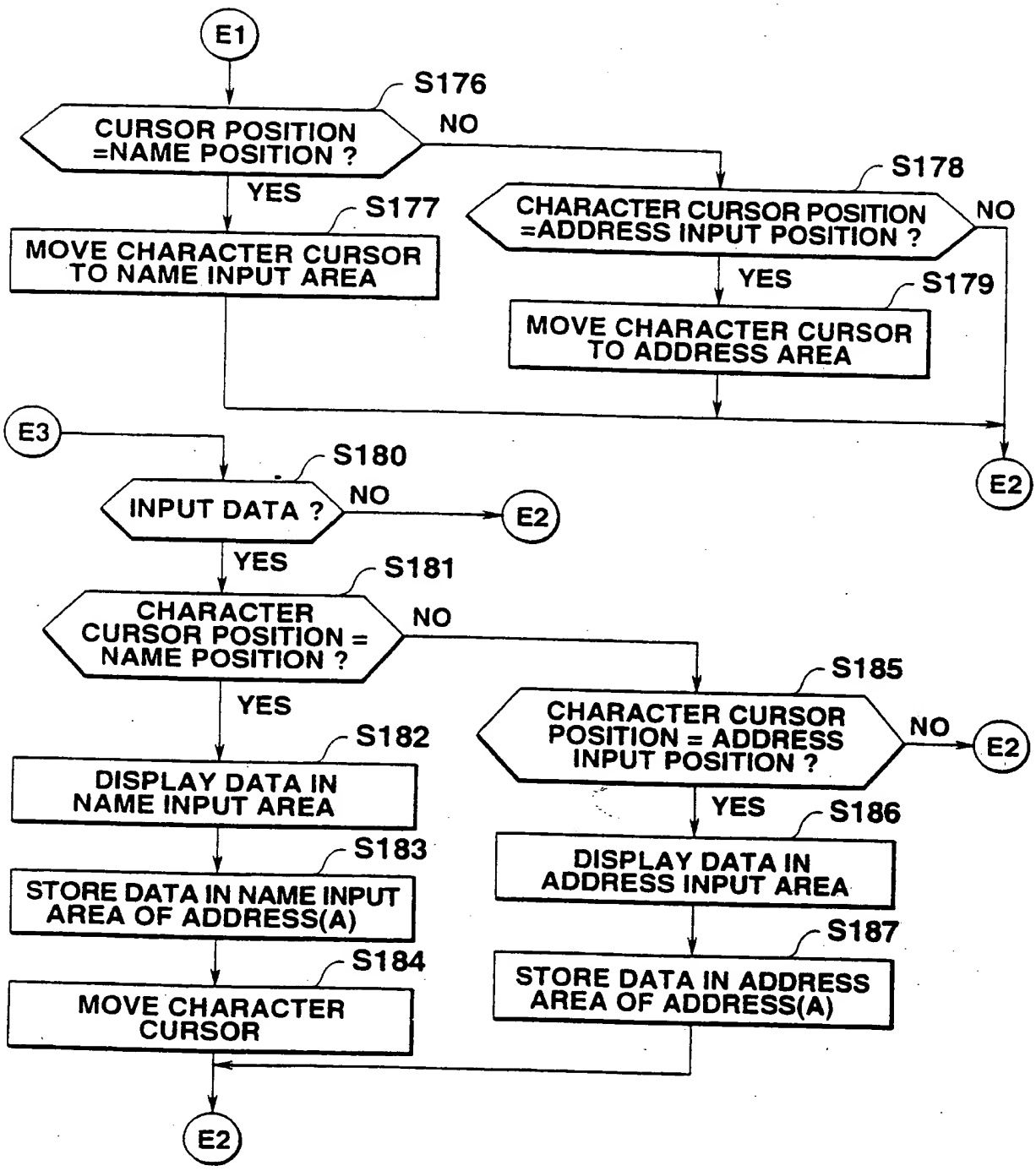

E1
S176
CURSOR POSITION =NAME POSITION ?
NO
YES
S177
MOVE CHARACTER CURSOR TO NAME INPUT AREA
S178
CHARACTER CURSOR POSITION =ADDRESS INPUT POSITION ?
NO
YES
S179
MOVE CHARACTER CURSOR TO ADDRESS AREA
E2
E3
S180
INPUT DATA ?
NO
E2
YES
S181
CHARACTER CURSOR POSITION = NAME POSITION ?
NO
YES
S182
DISPLAY DATA IN NAME INPUT AREA
S183
STORE DATA IN NAME INPUT AREA OF ADDRESS(A)
S184
MOVE CHARACTER CURSOR
E2
S185
CHARACTER CURSOR POSITION = ADDRESS INPUT POSITION ?
NO
E2
YES
S186
DISPLAY DATA IN ADDRESS INPUT AREA
S187
STORE DATA IN ADDRESS AREA OF ADDRESS(A)

FIG.23

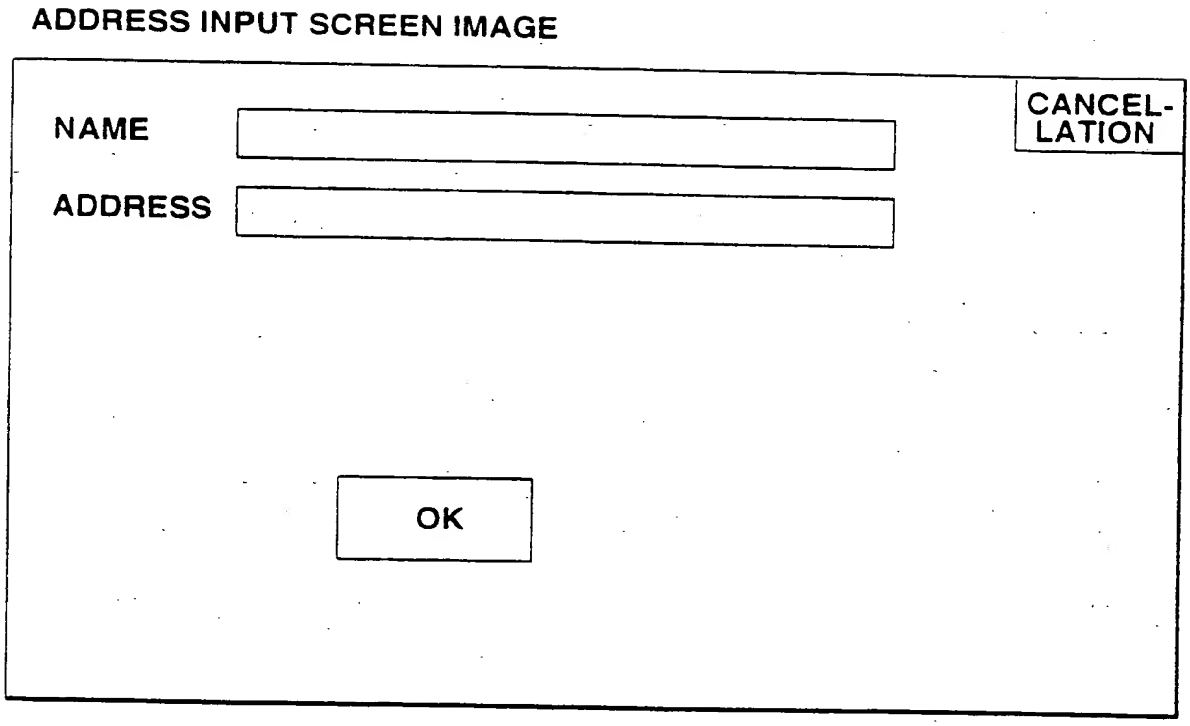


FIG.24

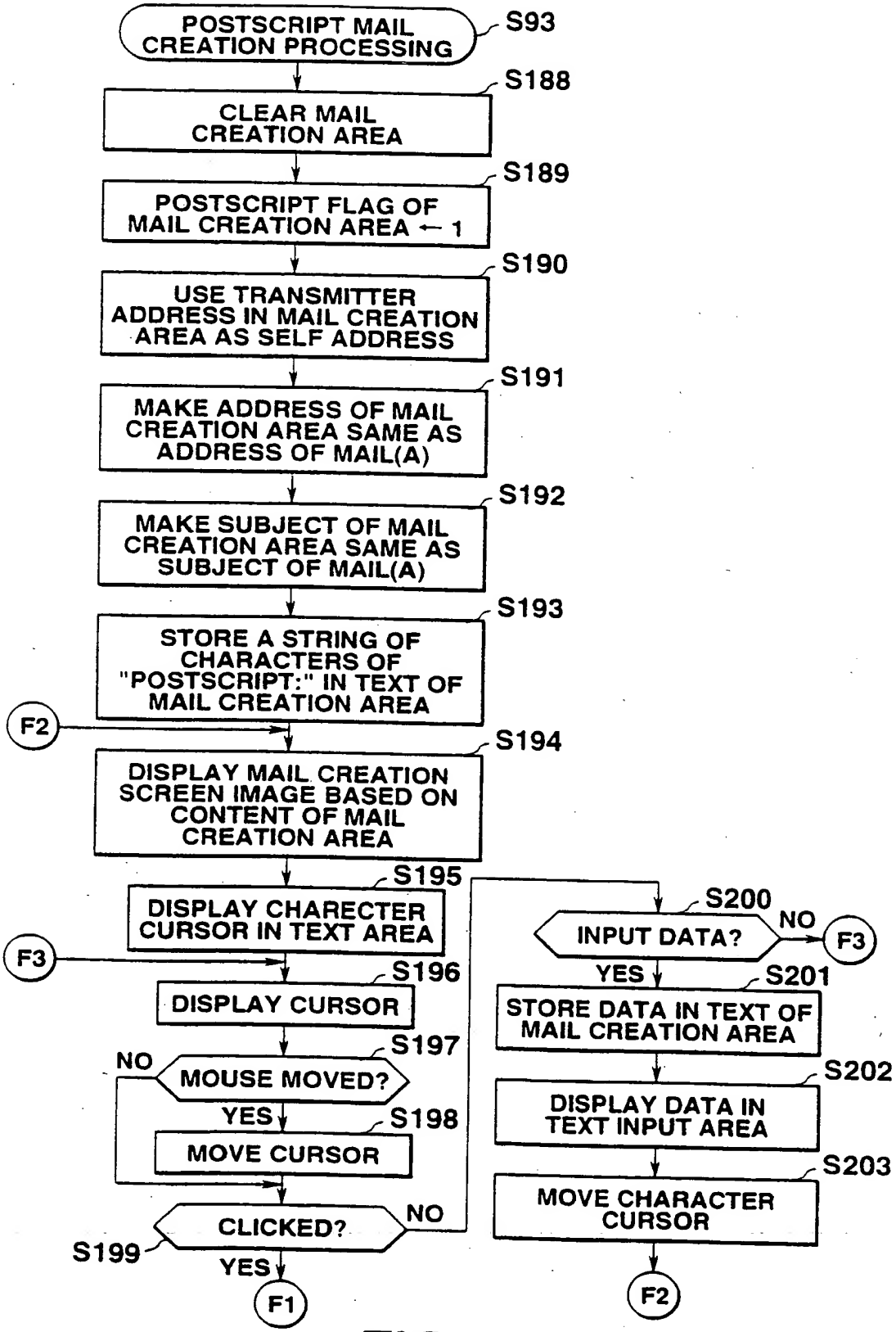

POSTSCRIPT MAIL CREATION PROCESSING
S93
S188
CLEAR MAIL CREATION AREA
S189
POSTSCRIPT FLAG OF MAIL CREATION AREA ← 1
S190
USE TRANSMITTER ADDRESS IN MAIL CREATION AREA AS SELF ADDRESS
S191
MAKE ADDRESS OF MAIL CREATION AREA SAME AS ADDRESS OF MAIL(A)
S192
MAKE SUBJECT OF MAIL CREATION AREA SAME AS SUBJECT OF MAIL(A)
S193
STORE A STRING OF CHARACTERS OF "POSTSCRIPT:" IN TEXT OF MAIL CREATION AREA
F2
S194
DISPLAY MAIL CREATION SCREEN IMAGE BASED ON CONTENT OF MAIL CREATION AREA
S195
DISPLAY CHARECTER CURSOR IN TEXT AREA
F3
S196
DISPLAY CURSOR
S197
NO
MOUSE MOVED?
YES
S198
MOVE CURSOR
NO
CLICKED?
S199
YES
F1
S200
INPUT DATA?
NO
F3
YES
S201
STORE DATA IN TEXT OF MAIL CREATION AREA
S202
DISPLAY DATA IN TEXT INPUT AREA
S203
MOVE CHARACTER CURSOR
F2

FIG.25

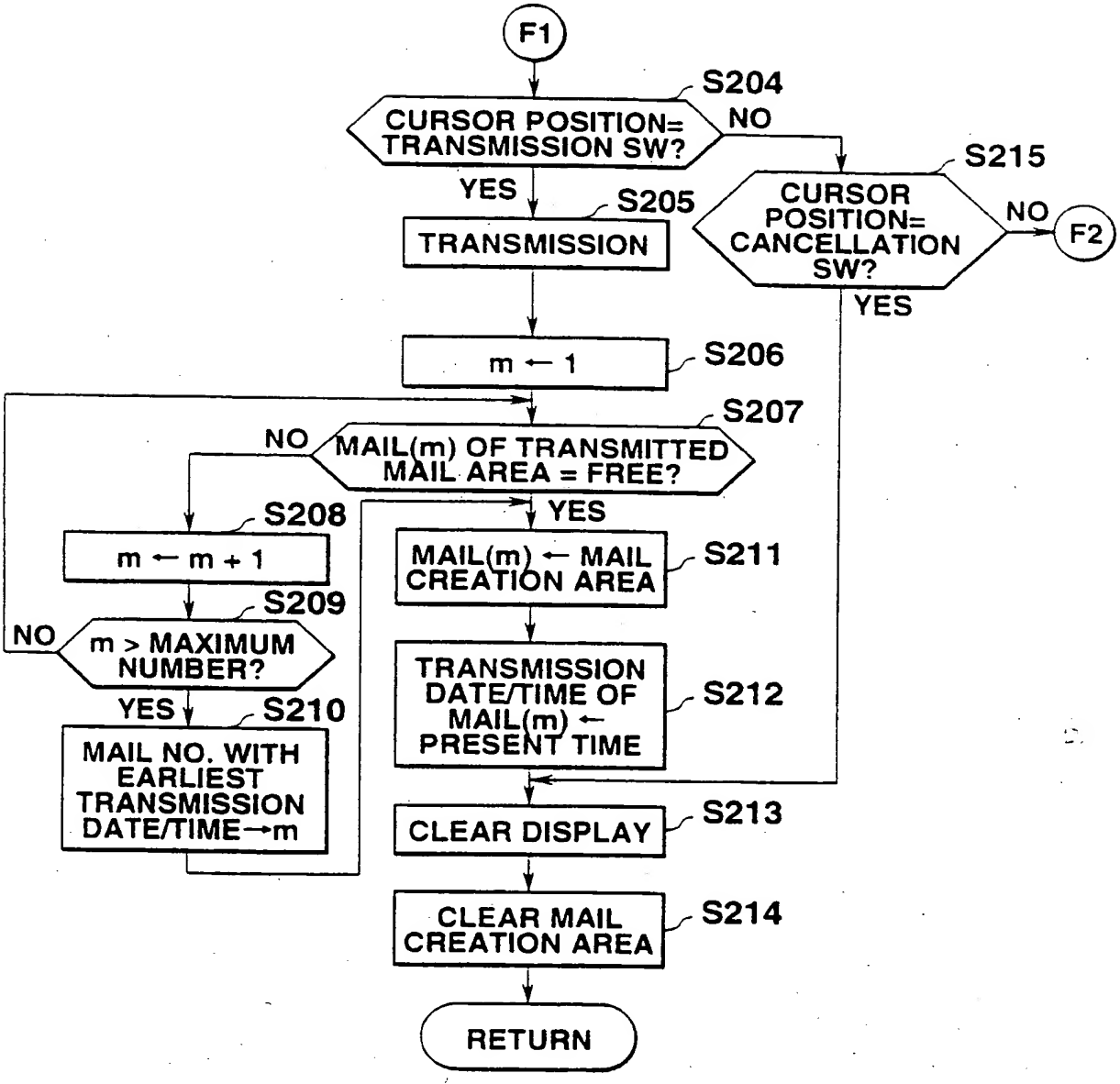


FIG.26

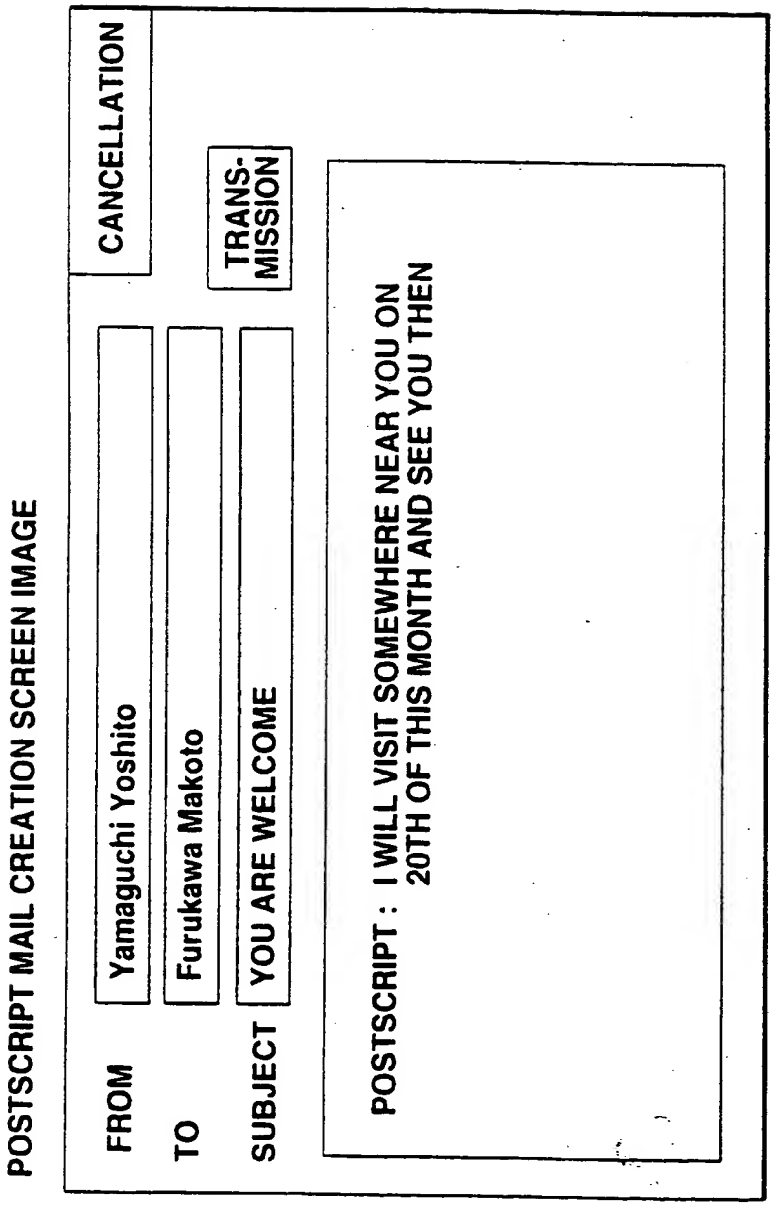

POSTSCRIPT MAIL CREATION SCREEN IMAGE
FROM
Yamaguchi Yoshito
CANCELLATION
TO
Furukawa Makoto
TRANS-MISSION
SUBJECT
YOU ARE WELCOME
POSTSCRIPT : I WILL VISIT SOMEWHERE NEAR YOU ON
20TH OF THIS MONTH AND SEE YOU THEN

FIG.27

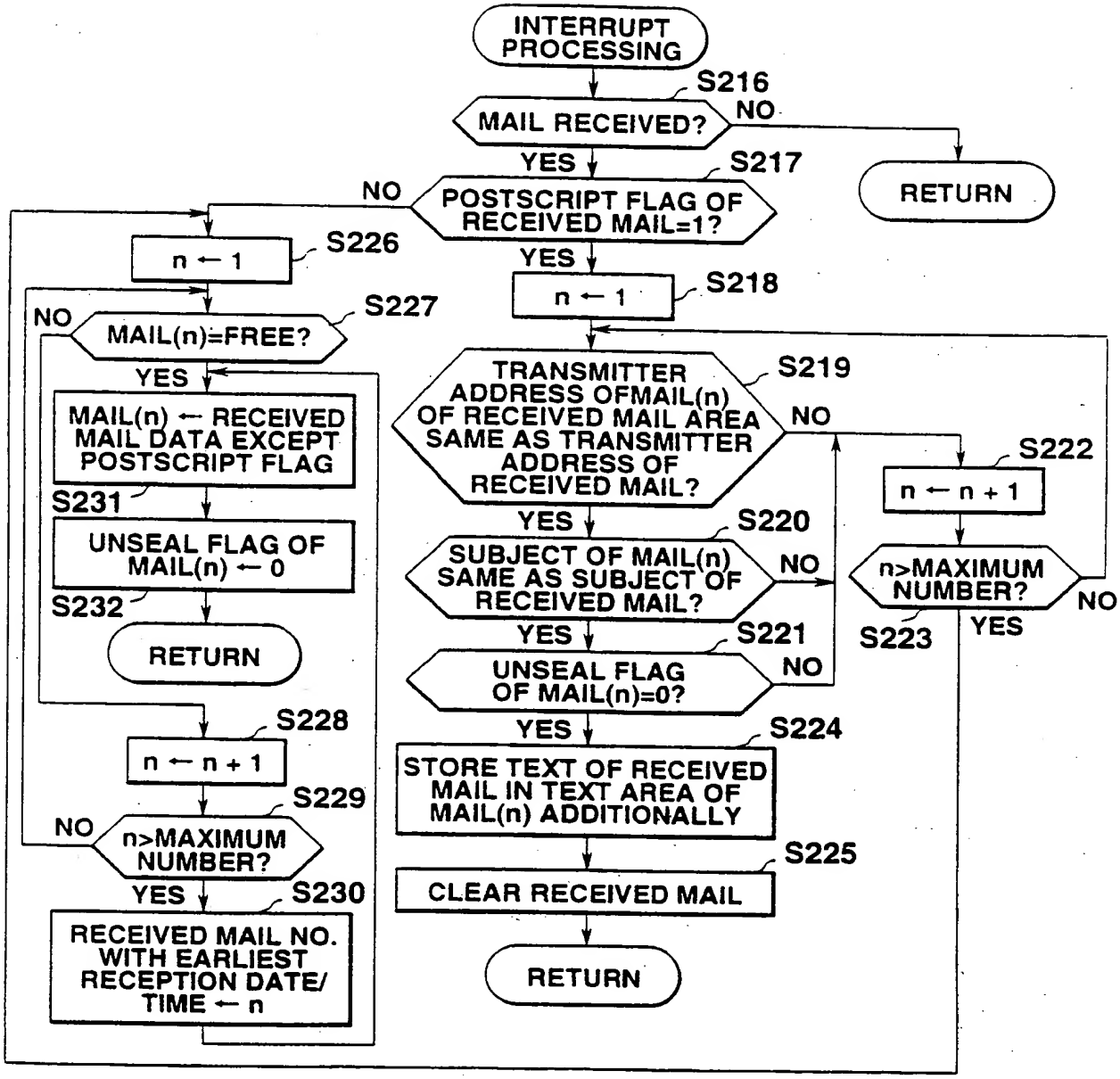


FIG.28

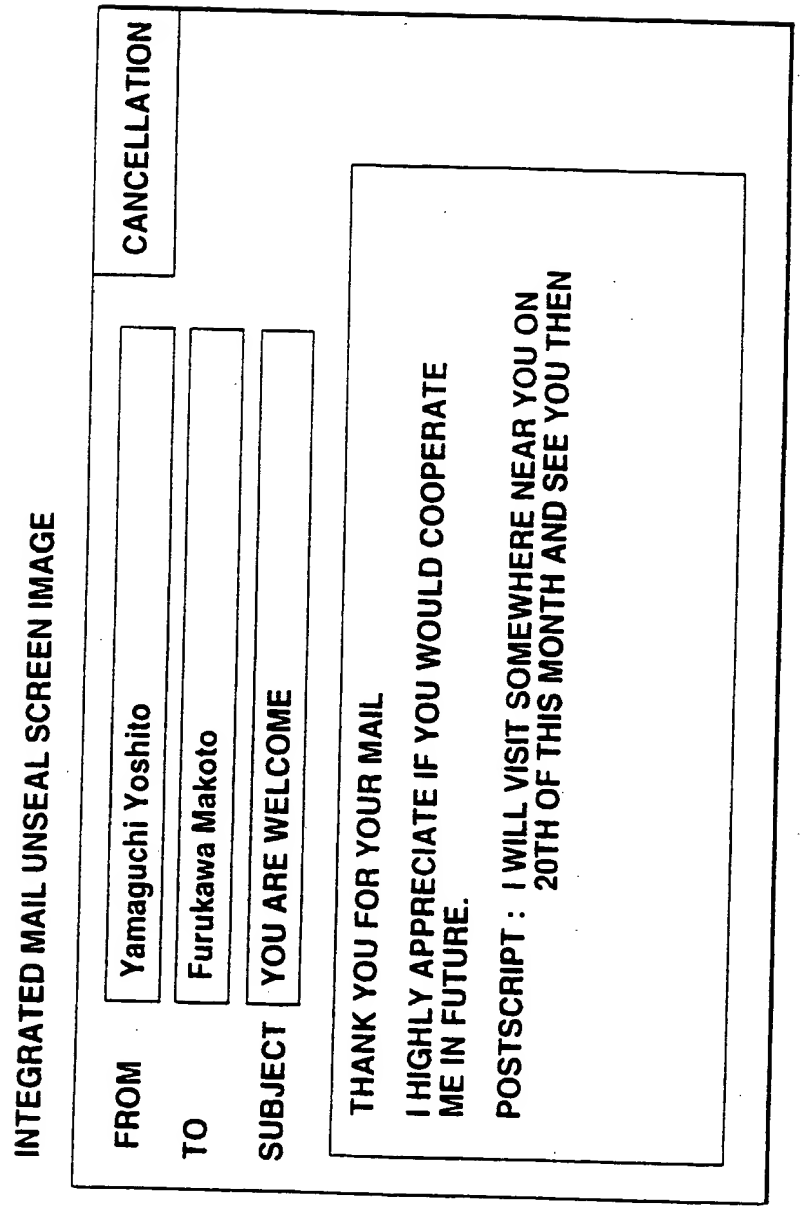

INTEGRATED MAIL UNSEAL SCREEN IMAGE
CANCELLATION
FROM
Yamaguchi Yoshito
TO
Furukawa Makoto
SUBJECT
YOU ARE WELCOME
THANK YOU FOR YOUR MAIL
I HIGHLY APPRECIATE IF YOU WOULD COOPERATE ME IN FUTURE.
POSTSCRIPT : I WILL VISIT SOMEWHERE NEAR YOU ON 20TH OF THIS MONTH AND SEE YOU THEN

FIG.29